

TUGAS AKHIR - SS141501

**PENERAPAN MODEL *HYBRID SINGULAR SPECTRUM ANALYSIS DEEP NEURAL NETWORK*
PADA PERAMALAN *INFLOW* DAN *OUTFLOW* UANG
KARTAL DI INDONESIA**

DIMAS EWIN ASHARI
NRP 062116 4500 0006

Dosen Pembimbing
Dr. Suhartono
M. Sjahid Akbar, S.Si., M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018



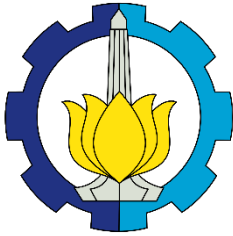
TUGAS AKHIR - SS141501

**PENERAPAN MODEL *HYBRID SINGULAR
SPECTRUM ANALYSIS DEEP NEURAL NETWORK*
PADA PERAMALAN *INFLOW* DAN *OUTFLOW*
UANG KARTAL DI INDONESIA**

**DIMAS EWIN ASHARI
NRP 062116 4500 0006**

**Dosen Pembimbing
Dr. Suhartono
M. Sjahid Akbar, S.Si., M.Si.**

**PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018**



FINAL PROJECT - SS141501

**HYBRID SINGULAR SPECTRUM ANALYSIS DEEP
NEURAL NETWORK MODEL FOR CURRENCY
FORECASTING IN INDONESIA**

**DIMAS EWIN ASHARI
SN 062116 4500 0006**

**Supervisors
Dr. Suhartono
M. Sjahid Akbar, S.Si., M.Si.**

**UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018**

LEMBAR PENGESAHAN

PENERAPAN MODEL *HYBRID SINGULAR SPECTRUM ANALYSIS DEEP NEURAL NETWORK* PADA PERAMALAN INFLOW DAN OUTFLOW UANG KARTAL DI INDONESIA

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
Pada
Program Studi Sarjana Departemen Statistika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

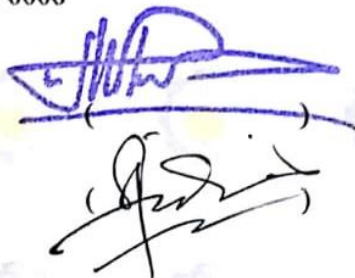
Oleh :

Dimas Ewin Ashari
NRP. 062116 4500 0006

Disetujui oleh Pembimbing:

Dr. Suhartono
NIP. 19710929 199512 1 001

M. Sjahid Akbar, S.Si., M.Si.
NIP. 19720705 199802 1 001



Mengetahui,
Kepala Departemen

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2018

**PENERAPAN MODEL *HYBRID SINGULAR*
SPECTRUM ANALYSIS DEEP NEURAL NETWORK
PADA PERAMALAN INFLOW DAN OUTFLOW
UANG KARTAL DI INDONESIA**

Nama Mahasiswa : Dimas Ewin Ashari
NRP : 062116 4500 0006
Departemen : Statistika
Dosen Pembimbing : Dr. Suhartono
M. Sjahid Akbar, S.Si., M.Si.

Abstrak

Perencanaan yang matang dalam penyusunan RKU oleh BI merupakan hal penting untuk menjaga ketersediaan uang di masyarakat. Salah satu faktor yang paling diperhitungkan dalam penyusunan RKU adalah inflow dan outflow. Penelitian ini bertujuan untuk mengetahui performa ARIMAX, DNN dan SSA-DNN dalam memprediksi inflow dan outflow uang kartal kertas di Indonesia dengan data per-bulan mulai Januari 2003 - Desember 2016. Prediksi inflow dan outflow yang akurat menjadi referensi penting agar RKU dapat tersusun secara optimal sehingga kebutuhan uang dapat terpenuhi dari sisi jumlah nominal, jenis komposisi pecahan yang sesuai, tepat waktu, serta layak edar. ARIMAX, DNN dan SSA-DNN akan dibandingkan berdasarkan RMSEP dan sMAPEP dan diharapkan mampu menangkap tren, musiman dan efek variasi kalender. Pada data simulasi dengan noise linier, ARIMAX memiliki akurasi terbaik, sedangkan pada data dengan noise non-linier, DNN lebih akurat dibandingkan ARIMAX. Sementara itu, pada data inflow dan outflow DNN adalah metode terbaik pada 13 dari 14 pecahan dibandingkan ARIMAX dan SSA-DNN.

Kata Kunci : ARIMAX, Deep Neural Network, Inflow, Outflow, Singular Spectrum Analysis, SSA-DNN, Variasi Kalender

(Halaman ini sengaja dikosongkan)

HYBRID SINGULAR SPECTRUM ANALYSIS DEEP NEURAL NETWORK MODEL FOR CURRENCY FORECASTING IN INDONESIA

Name : Dimas Ewin Ashari
Student Number : 062116 4500 0006
Department : Statistics
Supervisors : Dr. Suhartono
M. Sjahid Akbar, S.Si., M.Si.

Abstract

Mature planning in the preparation of RKU by BI is important to maintain the availability of money in the community. One of the most calculated factors in the preparation of RKU is the inflow and outflow. This study aims to determine the performance of ARIMAX, DNN and SSA-DNN in predicting inflow and outflow of currency in Indonesia with monthly data from January 2003 to December 2016. Accurate inflow and outflow prediction becomes an important reference so that RKU can be created optimally so that the need for money can be fulfilled in terms of nominal amount, the type of fractional composition, timely, and in proper condition. ARIMAX, DNN and SSA-DNN will be compared based on RMSEP and sMAPEP and are expected be able to capture the trend, seasonality and effects of calendar variations. DNN used in this research has 2 input types, DNN-1 uses significant PACF lag input from data & dummy while DNN-2 uses lag input according to AR order of ARIMAX. In the simulation data with linear noise, ARIMAX has the best accuracy, whereas in data with non-linear noise, DNN is more accurate than ARIMAX. Meanwhile, DNN application of inflow and outflow data are the best method on 13 of 14 fractions compared to ARIMAX and SSA-DNN, with 9 of them are DNN-1 type.

Keywords : ARIMAX, Currency, DNN, Singular Spectrum Analysis, SSA-DLNN, Calendar Variation

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang senantiasa memberikan rahmat, hidayah serta karunia-Nya. Sholawat serta salam semoga selalu tercurahkan untuk Nabi Muhammad SAW. Atas berkat rahmat Allah, penulis dapat menyelesaikan Tugas Akhir dengan berjudul **“Penerapan Model Hybrid Singular Spectrum Analysis Deep Neural Network pada Peramalan *Inflow* dan *Outflow* Uang Kartal di Indonesia”**. Dalam penyelesaian Tugas Akhir ini, penulis mendapatkan banyak bantuan dan dukungan dari berbagai pihak. Maka dari itu penulis dengan penuh hormat ingin mengucapkan terimakasih kepada:

1. Bapak Dr. Suhartono dan Bapak M. Sjahid Akbar, S.Si., M.Si. selaku dosen pembimbing yang telah banyak meluangkan waktu untuk memberikan arahan, bimbingan serta dukungan dalam penyusunan laporan Tugas Akhir ini.
2. Bapak Dr. rer. Pol. Dedy Dwi Prastyo, S.Si., M.Si. dan Dr. R. Mohamad Atok, S.Si., M.Si. selaku dosen penguji yang telah memberikan saran dan perbaikan Tugas Akhir ini.
3. Bapak Dr. Suhartono selaku Kepala Departemen Statistika ITS dan Bapak Dr. Sutikno, M.Si selaku Ketua Program Studi Sarjana Departemen Statistika FMIPA ITS.
4. Ibu Dra. Sri Mumpuni Retnaningsih, MT yang telah banyak memberikan dukungan selama penulis menempuh perkuliahan.
5. Seluruh bapak/ibu dosen pengajar di Jurusan Statistika ITS atas segala ilmu yang telah diberikan serta seluruh staf dan karyawan Jurusan Statistika atas kerja keras dan bantuannya selama ini.
6. Kedua orangtua saya, Hariyanto dan Mariyati serta kakak saya, Eka Maria Yuliantri, A.Md yang telah memberikan kasih sayang, ketulusan, dukungan dan doa.
7. Kepada teman-teman Lintas Jalur Statistika ITS 2016 yang telah berjuang bersama dan saling mendukung satu sama lain.

8. Pihak-pihak lain yang sudah banyak membantu dalam proses pengerjaan laporan Tugas Akhir ini yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih memiliki banyak kekurangan dan jauh dari kesempurnaan, maka segala kritik serta saran ataupun diskusi sangat dibutuhkan oleh penulis agar lebih baik kedepannya. Besar harapan penulis agar informasi sekecil apapun dari laporan ini dapat memberikan kebermanfaatan bagi berbagai pihak.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
COVER PAGE	iii
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xix
DAFTAR LAMPIRAN	xxi
 BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	7
1.3 Tujuan	7
1.4 Manfaat	8
1.5 Batasan Masalah	8
 BAB II TINJAUAN PUSTAKA	
2.1 Analisis Deret Waktu	9
2.2 Autoregressive Integrated Moving Average	9
2.3 ARIMAX	17
2.4 <i>Deep Learning</i>	18
2.5 <i>Singular Spectrum Analysis</i>	22
2.5.1 <i>Embedding</i>	23
2.5.2 <i>Singular Value Decomposition (SVD)</i>	24
2.5.3 <i>Eigentriple grouping</i>	24
2.5.4 <i>Diagonal averaging</i>	25
2.6 Model <i>Hybrid Singular Spectrum Analysis & Deep – Neural Network (SSA-DNN)</i>	26
2.7 Kriteria Evaluasi Model	28
2.8 Penelitian Sebelumnya.....	28
2.9 <i>White test</i>	30

2.10	<i>Inflow</i> dan <i>outflow</i> Uang Kartal	31
------	--	----

BAB III METODOLOGI PENELITIAN

3.1	Kajian simulasi	33
3.2	Kajian Terapan	37

BAB IV ANALISIS DAN PEMBAHASAN

4.1	Kajian Simulasi	45
4.1.1	Skenario 1 dengan <i>Noise</i> mengikuti AR(1)	47
4.1.2	Skenario 2 dengan <i>Noise</i> mengikuti ESTAR(1)	61
4.1.3	Perbandingan Performa ARIMAX, DNN dan SSA-DNN pada Data Simulasi	76
4.2	Karakteristik <i>inflow</i> dan <i>outflow</i> di Indonesia	81
4.3	Peramalan <i>inflow</i> dan <i>outflow</i> Menggunakan ARIMAX	85
4.4	Peramalan <i>inflow</i> dan <i>outflow</i> Menggunakan DNN-1	97
4.5	Peramalan <i>inflow</i> dan <i>outflow</i> Menggunakan DNN-2	104
4.6	Peramalan <i>inflow</i> dan <i>outflow</i> Menggunakan SSA-DNN	110
4.7	Perbandingan Model ARIMAX, DNN dan SSA-DNN	117
4.8	Peramalan Dengan Metode Terbaik	124

BAB V KESIMPULAN DAN SARAN.....127

5.1	Kesimpulan	127
5.2	Saran	128

DAFTAR PUSTAKA129

LAMPIRAN133

BIODATA PENULIS

DAFTAR GAMBAR

	Halaman
Gambar 1.1	Inflow Pecahan Rp.5000,-5
Gambar 2.1	Arsitektur FFNN 1 hidden layer 19
Gambar 2.2	Arsitektur Deep Neural Networks21
Gambar 2.3	Ilustrasi Metode Hybrid SSA-DNN.....27
Gambar 2.4	Ilustrasi Hubungan Stabilitas Sistem Keuangan dan Stabilitas Moneter28
Gambar 3.1	Tahapan ARIMAX42
Gambar 3.2	Tahapan SSA43
Gambar 3.3	Skema DNN.....43
Gambar 3.4	Diagram Alir44
Gambar 4.1	Pola Komponen Data Simulasi.....45
Gambar 4.2	Pola Komponen Noise Data Simulasi.....46
Gambar 4.3	Pola Data Simulasi Skenario 1 dan Skenario - 246
Gambar 4.4	Fits dan Forecast ARIMAX Y_{t-1-1}48
Gambar 4.5	Fits dan Forecast ARIMAX Skenario 1.....50
Gambar 4.6	PACF Y_{t-1-1}51
Gambar 4.7	Arsitektur DNN-1 Y_{t-1-1}52
Gambar 4.8	Fits dan Forecast DNN Y_{t-1-1}53
Gambar 4.9	Fits dan Forecast DNN Skenario 154
Gambar 4.10	Arsitektur DNN-2 Y_{t-1-1}55
Gambar 4.11	Fits dan Forecast DNN-2 Skenario 1.....57
Gambar 4.12	Plot Eigenvalues Y_{t-1-1}57
Gambar 4.13	Rekonstruksi Individu Y_{t-1-1}58
Gambar 4.14	Rekonstruksi & Pemodelan Agregat Y_{t-1-1} .58
Gambar 4.15	Arsitektur DNN Terbaik dari SSA-DNN Y_{t-1-1}59
Gambar 4.16	Fits dan Forecast SSA-DNN Y_{t-1-1}59
Gambar 4.17	Fits dan Forecast SSA-DNN Skenario 161
Gambar 4.18	ACF dan PACF Residual TSR Y_{t-2-1}62
Gambar 4.19	Fits dan Forecast ARIMAX Y_{t-2-1}63
Gambar 4.20	Fits dan Forecast ARIMAX Skenario 2.....64

Gambar 4.21	PACF Yt_2_1	65
Gambar 4.22	Arsitektur DNN Yt_2_1.....	66
Gambar 4.23	Fits dan Forecast DNN-1 Yt_2_1	67
Gambar 4.24	Fits dan Forecast DNN-1 Skenario 2.....	68
Gambar 4.25	Arsitektur DNN-2 Terbaik Yt_2_1	69
Gambar 4.26	Fits dan Forecast DNN-2 Skenario 2.....	70
Gambar 4.27	Plot Eigenvalues Yt_2_1	72
Gambar 4.28	Rekonstruksi Individu Yt_2_1.....	72
Gambar 4.29	Rekonstruksi & Pemodelan Agregat Yt_2_1.....	73
Gambar 4.30	Arsitektur DNN Terbaik dari SSA-DNN Yt_2_1	74
Gambar 4.31	Fits dan Forecast SSA-DNN Skenario 2.....	75
Gambar 4.32	Perbandingan RMSE Training & Testing - Simulasi	77
Gambar 4.33	Perbandingan sMAPE Training & Testing Simulasi Simulasi	78
Gambar 4.34	Perbandingan RMSEP Skenario 1 Menurut - Panjang Periode	80
Gambar 4.35	Perbandingan RMSEP Skenario 2 Menurut - Panjang Periode	81
Gambar 4.36	Perkembangan Outflow Nasional	82
Gambar 4.37	Perkembangan Inflow Nasional.....	83
Gambar 4.38	Aktual & Hasil Smoothing Data Outflow & Inflow	83
Gambar 4.39	Boxplot Outflow & Inflow Menurut Bulan ...	85
Gambar 4.40	ACF dan PACF Residual TSR Outflow Rp 100.000,-	87
Gambar 4.41	Fits & Forecast ARIMAX Outflow Rp100.000,-	88
Gambar 4.42	Fits & Forecast ARIMAX Outflow dan Inflow.....	94
Gambar 4.43	PACF Outflow Rp 100.000,-	98
Gambar 4.44	Arsitektur DNN Outflow Rp 100.000,-	98
Gambar 4.45	Fits & Forecast DNN-1 Outflow Rp 100.000,-	100

Gambar 4.46	Fits & Forecast DNN-1 Outflow dan Inflow	101
Gambar 4.47	Arsitektur DNN-2 Terbaik Outflow Rp100.000,-	105
Gambar 4.48	Fits & Forecast DNN-2 Outflow dan Inflow .	110
Gambar 4.49	Plot Eigenvalues Outflow Rp.100.000,-	110
Gambar 4.50	Rekonstruksi Individu Outflow Rp100.000,-	111
Gambar 4.51	Rekonstruksi & Pemodelan Agregat Rp100.000,-	111
Gambar 4.52	Arsitektur DNN dari SSA-DNN Outflow Rp100.000,-	112
Gambar 4.53	Fits & Forecast SSA-DNN Outflow dan Inflow	113
Gambar 4.54	sMAPE & sMAPEP Model ARIMAX, DNN-1, DNN-2 & SSA-DNN	118
Gambar 4.55	Rasio RMSE Model DNN-1, DNN-2 & SSA- DNN Terhadap ARIMAX	119
Gambar 4.56	Perbandingan RMSEP ARIMAX, DNN-1, DNN-2 dan SSA-DNN Data Outflow Menurut Periode Ramalan	120
Gambar 4.57	Perbandingan RMSEP ARIMAX, DNN-1, DNN-2 dan SSA-DNN Data Outflow Menurut Periode Ramalan	121
Gambar 4.58	Perbandingan Testing Aktual, ARIMAX, DNN-1, DNN-2 dan SSA-DNN	122
Gambar 4.59	Ringkasan Hasil Pemilihan Model Terbaik ...	123
Gambar 4.60	Ramalan 24 Periode Kedepan Berdasarkan Model Terbaik	124

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

	Halaman
Tabel 2.1	Karakteristik ACF dan PACF secara teoritis 11
Tabel 2.2	Penelitian Sebelumnya.....29
Tabel 3.1	Variabel Penelitian.....37
Tabel 3.2	Struktur Data.....38
Tabel 3.3	Variabel Dummy.....38
Tabel 4.1	TSR Yt ₁ ₁49
Tabel 4.2	Kemungkinan Model ARIMA Residual TSR Yt ₁ ₁50
Tabel 4.3	Estimasi Model ARIMA Residual TSR Yt ₁ ₁50
Tabel 4.4	Pemilihan Jumlah Neuron DNN Yt ₁ ₁53
Tabel 4.5	Model DNN-1 Terbaik Skenario 156
Tabel 4.6	Model DNN-2 Terbaik Skenario 158
Tabel 4.7	Model DNN Hybrid Terbaik Skenario 163
Tabel 4.8	TSR Yt ₂ ₁65
Tabel 4.9	Estimasi Model ARIMA Residual TSR Yt ₂ ₁66
Tabel 4.10	Pemilihan Jumlah Neuron DNN-1 Yt ₂ ₁69
Tabel 4.11	Model DNN-1 Terbaik Skenario 271
Tabel 4.12	Model DNN-2 Terbaik Skenario 274
Tabel 4.13	Model DNN Hybrid Terbaik Skenario 278
Tabel 4.14	Rasio Akurasi Data Training DNN dan SSA- DNN pada Data Simulasi terhadap ARIMAX...83
Tabel 4.15	Rasio Akurasi Data Testing DNN dan SSA- DNN pada Data Simulasi terhadap ARIMAX....83
Tabel 4.16	Statistika Deskriptif inflow dan outflow.....86
Tabel 4.17	Estimasi TSR Outflow Rp 100.000,-90
Tabel 4.18	Ljung-Box Residual TSR Outflow Rp 100.000,-91
Tabel 4.19	Uji ADF Residual TSR Outflow Rp 100.000,- ...91
Tabel 4.20	Kemungkinan Model ARIMA Residual TSR Outflow Rp 100.000,-92
Tabel 4.21	Model ARIMAX Outflow dan Inflow92
Tabel 4.22	RMSE dan sMAPE ARIMAX Kajian Terapan ..98

Tabel 4.23	Rasio RMSE dan sMAPE ARIMAX Testing Terhadap Training.....	99
Tabel 4.24	Model DNN-1 Outflow dan Inflow	106
Tabel 4.25	Rasio RMSE dan sMAPE DNN-1 Testing Terhadap Training.....	106
Tabel 4.26	Model DNN-2 Terbaik Outflow dan Inflow	112
Tabel 4.27	Model Terbaik SSA-DNN Inflow dan Outflow ..	119
Tabel 4.28	RMSEP Outflow Model ARIMAX, DNN-1, DNN-2 & SSA-DNN	123
Tabel 4.29	RMSEP Inflow Model ARIMAX, DNN-1, DNN-2 & SSA-DNN	123

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Data Inflow & Outflow	133
Lampiran 2 ACF & PACF Residual TSR Skenario 1	134
Lampiran 3 ACF & PACF Residual TSR Skenario 2	135
Lampiran 4 PACF Data Simulasi untuk input DNN-1	136
Lampiran 5 Akurasi Data Simulasi.....	137
Lampiran 6 Parameter ARIMAX Outflow Rp.100.000,-	138
Lampiran 7 Parameter ARIMAX Outflow Rp50.000,-	139
Lampiran 8 Parameter ARIMAX Outflow Rp.20.000,-	140
Lampiran 9 Parameter ARIMAX Outflow Rp.10.000,-	141
Lampiran 10 Parameter ARIMAX Outflow Rp.5.000,-	142
Lampiran 11 Parameter ARIMAX Outflow Rp.2.000,-	143
Lampiran 12 Parameter ARIMAX Outflow Rp.1.000,	144
Lampiran 13 Parameter ARIMAX Inflow Rp.100.000,-	145
Lampiran 14 Parameter ARIMAX Inflow Rp.50.000,-	146
Lampiran 15 Parameter ARIMAX Inflow Rp.20.000,-	147
Lampiran 16 Parameter ARIMAX Inflow Rp.10.000,-	148
Lampiran 17 Parameter ARIMAX Inflow Rp.5.000,-	149
Lampiran 18 Parameter ARIMAX Inflow Rp.2.000,-	150
Lampiran 19 Parameter ARIMAX Inflow Rp.1.000,-	151
Lampiran 20 <i>P-value</i> Uji Linieritas (White Test) Input DNN-1	152
Lampiran 21 DNN-1 Outflow Rp 100.000,-	155
Lampiran 22 DNN-1 Outflow Rp 50.000,-	158
Lampiran 23 DNN-1 Outflow Rp 20.000,-	157
Lampiran 24 DNN-1 Outflow Rp 10.000,-	158
Lampiran 25 DNN-1 Outflow Rp 5.000,-	159
Lampiran 26 DNN-1 Outflow Rp.2.000,-	160
Lampiran 27 DNN-1 Outflow Rp 1.000,-	161
Lampiran 28 DNN-1 Inflow Rp 100.000,-	162
Lampiran 29 DNN-1 Inflow Rp 50.000,-	163
Lampiran 30 DNN-1 Inflow Rp 20.000,-	164
Lampiran 31 DNN-1 Inflow Rp 10.000,-	165

Lampiran 32	DNN-1 Inflow Rp 5.000,-	166
Lampiran 33	DNN-1 Inflow Rp.2.000,-	167
Lampiran 34	DNN-1 Inflow Rp 1.000,-	168
Lampiran 35	<i>P-value</i> Uji Linieritas (White Test) Input DNN-2	169
Lampiran 36	DNN-2 Outflow Rp 100.000,-	171
Lampiran 37	DNN-2 Outflow Rp 50.000,-	172
Lampiran 38	DNN-2 Outflow Rp 20.000,-	173
Lampiran 39	DNN-2 Outflow Rp 10.000,-	174
Lampiran 40	DNN-2 Outflow Rp 5.000,-	175
Lampiran 41	DNN-2 Outflow Rp 2.000,-	176
Lampiran 42	DNN-2 Outflow Rp 1.000,-	177
Lampiran 43	DNN-2 Inflow Rp 100.000,-	178
Lampiran 44	DNN-2 Inflow Rp 50.000,-	179
Lampiran 45	DNN-2 Inflow Rp 20.000,-	180
Lampiran 46	DNN-2 Inflow Rp 10.000,-	181
Lampiran 47	DNN-2 Inflow Rp 5.000,-	182
Lampiran 48	DNN-2 Inflow Rp 2.000,-	183
Lampiran 49	DNN-2 Inflow Rp 1.000,-	184
Lampiran 50	<i>ADF Test & White Test</i> Input DNN dari SSA - DNN	185
Lampiran 51	SSA-DNN Outflow Rp 100.000,-	187
Lampiran 52	SSA-DNN Outflow Rp 50.000,-	190
Lampiran 53	SSA-DNN Outflow Rp 20.000,-	193
Lampiran 54	SSA-DNN Outflow Rp 10.000,-	196
Lampiran 55	SSA-DNN Outflow Rp 5.000,-	199
Lampiran 56	SSA-DNN Outflow Rp 2.000,-	202
Lampiran 57	SSA-DNN Outflow Rp 1.000,-	205
Lampiran 58	SSA-DNN Inflow Rp 100.000,-	208
Lampiran 59	SSA-DNN Inflow Rp 50.000,-	211
Lampiran 60	SSA-DNN Inflow Rp 20.000,-	214
Lampiran 61	SSA-DNN Inflow Rp 10.000,-	217
Lampiran 62	SSA-DNN Inflow Rp 5.000,-	220
Lampiran 63	SSA-DNN Inflow Rp 2.000,-	223
Lampiran 64	SSA-DNN Inflow Rp 1.000,-	226

Lampiran 65	Weights DNN Outflow Rp 100.000,-.....	229
Lampiran 66	Weights DNN Outflow Rp 50.000,-.....	230
Lampiran 67	Weights DNN Outflow Rp 20.000,-.....	231
Lampiran 68	Weights DNN Outflow Rp 10.000,-.....	232
Lampiran 69	Weights DNN Outflow Rp 5.000,-.....	233
Lampiran 70	Weights DNN Outflow Rp 2.000,-.....	235
Lampiran 71	Weights DNN Outflow Rp 1.000,-.....	237
Lampiran 72	Weights DNN Inflow Rp 100.000,-	238
Lampiran 73	Weights DNN Inflow Rp 50.000,-	240
Lampiran 74	Weights DNN Inflow Rp 20.000,-	241
Lampiran 75	Weights DNN Inflow Rp 10.000,-	242
Lampiran 76	Weights DNN Inflow Rp 5.000,-	243
Lampiran 77	Weights DNN Inflow Rp 2.000,-	243
Lampiran 78	Weights DNN Inflow Rp 1.000,-	245
Lampiran 79	Model Ramalan Manual ARIMAX <i>Outflow</i> Rp 100.000,-	247
Lampiran 80	Uji Non-Linieritas (<i>White Test</i>) Secara Simultan dengan Lag Masing-masing Series	248
Lampiran 81	Syntax-SAS ARIMAX <i>Outflow</i> Rp 100.000,-	249
Lampiran 82	Syntax-R DNN <i>Outflow</i> Rp 100.000,-	250
Lampiran 83	Syntax-R SSA-DNN <i>Outflow</i> Rp 100.000,- ...	253

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Uang kartal masih menjadi alat bayar yang sah dan wajib digunakan oleh masyarakat dalam melakukan transaksi jual beli sehari-hari. Uang mengambil peranan yang sangat vital dalam kehidupan dan berkaitan erat dengan kondisi perekonomian dan stabilitas harga barang. Harga suatu barang merupakan interaksi antara jumlah uang yang beredar dengan jumlah uang yang tersedia di pasar, sehingga penting untuk mengatur keseimbangan jumlah uang yang beredar dalam menjaga stabilitas harga dan stabilitas pertumbuhan ekonomi melalui kebijakan moneter (Ambarini, 2015).

Bank Indonesia selaku bank sentral Republik Indonesia mengemban misi dalam hal pengedaran uang untuk memenuhi kebutuhan uang di masyarakat dalam jumlah nominal yang cukup, jenis pecahan yang sesuai, tepat waktu dan dalam kondisi layak edar. Uang kartal disirkulasikan ke seluruh wilayah negara sesuai dengan kebutuhan gerak roda perekonomian tiap wilayah. Bila kebutuhan uang beredar tidak tercukupi maka perekonomian negara akan terganggu, tidak stabil dan bahkan bisa kolaps. Sedangkan jika jumlah uang melebihi kebutuhan, nilai mata uang akan turun. Bank Indonesia menyusun Rencana Kebutuhan Uang (RKU) untuk memperkirakan kebutuhan tambahan uang yang mencakup jumlah dan komposisi uang baik di Kantor Pusat BI (KPBI) maupun Kantor BI (KBI) di berbagai daerah. Pertimbangan utama penyusunan RKU adalah aliran uang masuk (*inflow*) dan keluar (*outflow*), posisi kas yang tersedia, jumlah uang yang dimusnahkan, kas minimum yang harus tersedia serta kondisi ekonomi dan geografis (Bank Indonesia, 2011).

Faktor *inflow* dan *outflow* merupakan cerminan pergerakan permintaan uang kartal yang dipengaruhi oleh pertumbuhan ekonomi daerah, angka inflasi, perbandingan jumlah kredit dan dana pihak ketiga di perbankan, faktor musiman (musiman bulan

dan musiman lain seperti Hari Raya Idul Fitri yang menyebabkan adanya efek variasi kalender) hingga sosial budaya (Bank Indonesia, 2011). Prediksi kebutuhan uang yang tepat untuk setiap komposisi pecahan sangat bermanfaat bagi BI dalam menyusun RKU, sehingga telah banyak penelitian yang dilakukan terkait peramalan *inflow* dan *outflow* menggunakan berbagai macam metode. Umumnya perubahan nilai *inflow* dan *outflow* mengandung pola *trend* maupun musiman, oleh karena itu untuk mengetahui komponennya secara terpisah dapat dilakukan dekomposisi. Dekomposisi pola data deret waktu menjadi subpola tren, musiman dan *noise* dapat mempermudah proses analisis deret waktu dalam menyusun ramalan (Bowerman & O'Connell, 1993).

Guler dan Talasli (2010) memodelkan peredaran uang di Turki menggunakan *Autoregressive Moving Average – Generalized Autoregressive Conditional Heteroscedasticity* (ARMA-GARCH) dengan memperhatikan pola musiman hari, minggu, bulan dan hari libur. Model ARIMA dengan variabel *dummy* musiman diestimasi menggunakan *non-linear least square* menghasilkan residual tidak homogen (terjadi heteroskedastisitas) sehingga digunakan GARCH. Rachmawati (2015) menggunakan model regresi time series, ARIMA dan ARIMAX untuk meramalkan *inflow* dan *outflow* yang menghasilkan varians residual tidak homogen. Saputri (2016) meramalkan *inflow* dan *outflow* di Indonesia menggunakan model *hybrid Quantil Regression Neural Network* (QRNN) yang dapat menangkap pola heteroskedastisitas dan non-linieritas. Peramalan interval QRNN pada penelitian tersebut menunjukkan adanya *crossing* antar kuantil karena estimasinya dilakukan secara independen untuk batas atas dan batas bawah selang kepercayaan. Salehah (2017) meramalkan *inflow-outflow* Jawa Timur dengan ARIMAX-QR. Setyowati (2017) meramalkan *inflow* dan *outflow* di Indonesia menggunakan model *hybrid Singular Spectrum Analysis Neural Network* (SSA-NN) yang dapat menangkap pola tren dan musiman pada data *inflow-outflow* namun kurang mampu menangkap efek variasi kalender. Penelitian tersebut memodelkan komponen dari SSA

secara individu maupun agregat menggunakan DNN baik untuk komponen tren, musiman maupun *noise*. Metode lain yang saat ini sedang populer di bidang analisis data adalah model-model yang melibatkan *Artificial Intelligence* dalam versi yang relatif sederhana.

Artificial Intelligence (AI) dipelopori oleh Stuart Russel dan Peter Norvig bersama buku berjudul *Artificial Intelligence: A Modern Approach* kemudian menjadi sangat populer dan banyak digunakan pada beberapa tahun terakhir. AI memiliki cakupan yang sangat luas salah satu yang terkenal dalam analisis data dan kalangan *data scientist* adalah bidang *machine learning*. Pada awal tahun 2006, *deep learning* mulai diperkenalkan secara luas sebagai bagian dari *machine learning* dan banyak digunakan untuk klasifikasi, *forecasting* dan berbagai analisis data lain (Patterson & Gibson, 2017). Apabila terdapat permasalahan nonlinieritas, maka model *neural network* dapat digunakan. Dalam peramalan, metode *neural network* yang banyak digunakan adalah *Feedforward Neural Network* (FFNN). Pada FFNN, proses dimulai dari *input* yang diterima oleh *neuron*, dimana *neuron-neuron* ini dikelompokkan dalam *layer*. Informasi yang diterima dari *layer input* dilanjutkan ke *layer-layer* dalam FFNN secara berurutan hingga mencapai *layer output*. *Layer* yang berbeda diantara *input* dan *output* disebut *hidden layer* (Suhartono, 2007). FFNN umumnya memiliki 1 *hidden layer* dalam arsitekturnya, namun juga dapat digunakan lebih dari satu *hidden layer*, penggunaan multi *hidden layer* dalam FFNN termasuk dalam *deep learning* dan selanjutnya dalam penelitian ini akan disebut *deep neural network* (DNN).

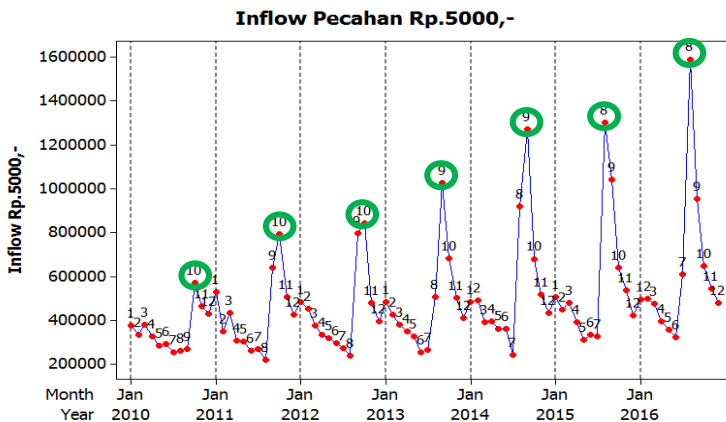
Busseti (2012) mengimplementasikan arsitektur *deep learning* untuk meramalkan *power loads* dan menemukan bahwa *deep learning* menghasilkan hasil yang sangat bagus untuk regresi linier dan kernel. Pada dataset yang besar, *deep learning* mampu diterapkan pada model non-linier yang kompleks tanpa menemui masalah *over-fitting* yang berarti. *Deep learning* memungkinkan untuk menambahkan kerumitan yang signifikan pada model tanpa

menentukan variasi bentuk yang harus diambil sehingga banyak digunakan dalam berbagai penelitian di banyak negara. Sedangkan *Singular Spectrum Analysis* (SSA) pernah digunakan oleh Kumar & Jain (2010) dalam memodelkan dan *forecasting* konsumsi energi di India. SSA menyediakan estimasi dimensi statistik. SSA juga menggambarkan fenomena fisik utama yang tercermin dari data. Hal ini memberikan filter spektral adaptif yang terkait dengan osilasi sistem yang dominan dan mengklarifikasi karakteristik noise dari data (Vautard & Ghil, 1989). Yu, Li & Zhang (2017) menggunakan *SSA-Elman Neural Network* yang di-hybrid-kan dengan 3 metode yaitu EMD, EEMD & CEEMDN untuk meramalkan kecepatan angin. Hasilnya menunjukkan bahwa setelah SSA digunakan, kinerja model hibrida baru yang mereka usulkan membaik secara signifikan. SSA menangani komponen dengan frekuensi tertinggi yang terpilah dari teknologi dekomposisi.

X. Zhang, Wang & K. Zhang (2017) meramalkan beban listrik jangka pendek menggunakan SSA-SVM dengan optimasi *Cuckoo Search* (CS). Data empiris didekomposisi dengan SSA kemudian dimodelkan dengan SVM dan optimasi CS. Kesimpulan dari penelitian tersebut adalah bahwa SSA lebih kontributif untuk memisahkan komponen *noise* dalam meningkatkan kinerja SVM dibanding CS dan model CS-SSA-SVM adalah model terbaik dibandingkan dengan model SVM, SSA-SVM, CS-SVM, SARIMA dan BPNN. Liu, Mi & Li (2018) menggunakan *SSA-Extreme Learning Machine* (ELM) dengan *Variational Mode Decomposition* (VMD) dan *Long Short Term Memory* (LSTM) untuk meramalakan kecepatan angin di Cina. VMD digunakan untuk menguraikan data kecepatan angin yang asli menjadi *sublayer*, SSA digunakan untuk mengekstrak lebih lanjut informasi tren dari semua *sublayer*, LSTM digunakan untuk menyelesaikan peramalan dari *sublayer* berfrekuensi rendah yang diperoleh dari VMD-SSA sedangkan ELM digunakan untuk menyelesaikan peramalan *sublayer* berfrekuensi tinggi yang diperoleh dari VMD-SSA. Hasilnya menunjukkan bahwa metode hibrida VMD-SSA-LSTM-ELM adalah model terbaik dibandingkan dengan ARIMA,

ELM, LSTM, VMD-ELM, dan VMD-LSTM-ELM yang tidak menggunakan SSA. Model VMD-SSA-LSTM-ELM dinilai lebih efektif dan *robust* dalam mengekstrak informasi tren.

Penggunaan *neural network* dalam penelitian ini dilatar belakangi oleh hasil penelitian Saputri (2016) yang menyimpulkan bahwa terdapat pola non-linier pada *inflow* semua pecahan mata uang kartal kertas dan 4 dari 7 pecahan *outflow* menurut data tahun 2003 hingga 2015 sehingga *neural network* digunakan untuk menangkap pola non-linier ini. FFNN adalah *neural network* yang sering digunakan dalam *time series*, umumnya memiliki 1 lapis *hidden layer*. Sedangkan DNN yang akan digunakan dalam penelitian ini adalah seperti FFNN namun menggunakan 2 *hidden layer* sehingga pemrosesan data diharapkan lebih sensitif dalam menangkap pola deret waktu. Beberapa sumber referensi menyebut FFNN *multilayer* ini sebagai *deep feedforward networks* atau *multilayer perceptrons* (MLPs). Gambar 1.1 adalah plot perkembangan *inflow* pecahan Rp.5000,- yang dipilih sebagai contoh untuk menggambarkan adanya tren, musiman, efek variasi kalender dan *outlier* pada data *inflow* serta *outflow* uang kartal kertas di Indonesia. Pecahan Rp.5000,- dipilih karena dianggap paling banyak digunakan pada perayaan Hari Raya Idul Fitri.



Gambar 1.1 *Inflow* Pecahan Rp.5000,-

Gambar 1.1 menunjukkan bahwa terjadi pergeseran *inflow* tertinggi setiap tahun dari bulan ke-10 (Oktober), bulan ke-9 (September) ke bulan Agustus. Hal ini bertepatan dengan Hari Raya Idul Fitri pada 7 tahun terakhir yang terjadi pada 3 bulan tersebut, sehingga ada indikasi bahwa terjadi efek variasi kalender pada arus *inflow* di Indonesia. Adanya tren naik juga dapat dilihat dari plot walaupun kenaikannya tidak terlalu jelas, sedangkan efek musiman dapat diketahui dari *inflow* yang cenderung rendah di bulan Januari sampai Juli dan cenderung lebih tinggi pada akhir tahun. Hal ini menunjukkan bahwa terdapat beberapa pola dalam arus uang kartal di Indonesia sehingga metode dekomposisi dengan SSA diharapkan dapat memisahkan pola-pola tersebut kedalam komponen baru dan juga *noise*. Oleh karena itu, penelitian ini akan membahas tentang penerapan model *hybrid* SSA-DNN yang diharapkan dapat menghasilkan ramalan dengan akurasi lebih tinggi, dan dapat menangkap pola *trends*, musiman, *outliers*, efek variasi kalender dan pola non-linier.

Perbedaan penelitian ini dengan penelitian Setyowati (2017) adalah pada pemodelan *hybrid* DNN dari hasil SSA. Penelitian tersebut yang menggunakan DNN pada komponen tren, musiman dan *noise* secara individu maupun agregat, sedangkan penelitian ini akan menggunakan *trend analysis* untuk komponen tren, regresi *dummy* untuk komponen musiman dan DNN untuk komponen *noise* yang dihasilkan dari SSA. Selain itu *hidden layer* yang digunakan dalam penelitian ini adalah 2 *hidden layer* sedangkan penelitian sebelumnya menggunakan FFNN 1 *hidden layer* pada hasil dekomposisi SSA. Dalam penelitian ini, performansi SSA-DNN akan dibandingkan dengan ARIMAX yang dikenal mampu menangkap efek variasi kalender dan DNN yang bisa menangkap pola non-linier. Evaluasi model dilakukan dengan kriteria *root mean square error prediction* (RMSEP) dan *symmetric mean absolute percentage error* (SMAPE) pada data bangkitan sebagai kajian simulasi dan data *inflow* dan *outflow* sebagai kajian terapan. Data *inflow-outflow* yang digunakan diambil dalam periode bulanan mulai dari Januari 2003 hingga Desember 2017.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, permasalahan utama yang dibahas dalam penelitian ini adalah bagaimana perbandingan performa dalam model *hybrid* SSA-DNN dengan model ARIMAX dan DNN dalam memodelkan *inflow* dan *outflow* di Indonesia yang mengandung tren, musiman, *outliers* dan efek variasi kalender. Sedangkan detail permasalahan yang dibahas yang akan dibahas adalah sebagai berikut:

1. Bagaimana pemodelan *inflow* dan *outflow* di Indoneasia menggunakan model ARIMAX?
2. Bagaimana pemodelan *inflow* dan *outflow* di Indonesia menggunakan DNN?
3. Bagaimana pemodelan *inflow* dan *outflow* di Indonesia menggunakan SSA-DNN?
4. Bagaimana perbandingan performa dan akurasi hasil peramalan *inflow* dan *outflow* menggunakan metode ARIMAX, DNN dan *hybrid* SSA-DNN?
5. Bagaimana hasil peramalan pada data *inflow* dan *outflow* di Indonesia pada periode 2018 dan 2019?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah disebutkan, tujuan utama yang ingin dicapai dalam penelitian ini adalah membandingkan performa model *hybrid* SSA-DNN dengan model ARIMAX dan DNN dalam meramalkan *inflow* dan *outflow* di Indonesia yang mengandung tren, musiman, *outliers* dan efek variasi kalender. Sedangkan detail tujuan yang ingin dicapai adalah sebagai berikut:

1. Memodelkan data *inflow* dan *outflow* di Indonesia dengan model ARIMAX.
2. Memodelkan data *inflow* dan *outflow* di Indonesia dengan model DNN.
3. Memodelkan data *inflow* dan *outflow* di Indonesia dengan model SSA-DNN.

4. Membandingkan performa dan akurasi hasil peramalan *inflow* dan *outflow* menggunakan metode ARIMAX, DNN dan *hybrid* SSA-DNN.
5. Meramalkan *inflow* dan *outflow* di Indonesia pada periode 2018 dan 2019.

1.4 Manfaat

Manfaat yang diharapkan dapat diberikan melalui penelitian ini diantaranya sebagai berikut:

1. Memberikan informasi yang dapat dijadikan sebagai referensi pertimbangan bagi Bank Indonesia dalam percetakan dan/atau pengadaan uang kartal tahun 2018 sehingga kebutuhan uang di masyarakat dapat dipenuhi dari segi jumlah nominal dan jenis pecahan serta stabilitas jumlah uang beredar dapat terjaga.
2. Memberikan wawasan keilmuan statistika terkait dengan penerapan metode ARIMAX, DNN serta *hybrid singular spectrum analysis deep neural network* (SSA-DNN) dalam memodelkan deret waktu yang mengandung tren, musiman, *outliers* dan variasi kalender.

1.5 Batasan Masalah

Penelitian ini membatasi peramalan *inflow* dan *outflow* uang kartal yang masuk dan keluar Bank Indonesia secara nasional dan khusus untuk uang kartal kertas per pecahan yaitu pecahan 1000, 2000, 5000, 10000, 20000, 50000 dan 100000 rupiah. *Deep learning* yang digunakan dalam penelitian ini adalah *deep learning feedforward neural network* dengan 2 *hidden layer* yang selanjutnya akan disebut sebagai DNN. Metode yang digunakan adalah model *hybrid* SSA-DNN, ARIMAX dan DNN. Variabel *dummy* yang digunakan hanya *dummy* untuk musiman bulan kalender masehi, tren, *outliers* dan variasi kalender khusus untuk efek hari raya Idul Fitri.

BAB II

TINJAUAN PUSTAKA

Tinjauan pustaka dalam penelitian ini memuat analisis deret waktu, model ARIMA, model ARIMAX, *deep learning* untuk *time series*, SSA, model *hybrid* SSA-DNN, kriteria evaluasi model untuk membandingkan kebaikan antar metode, serta *inflow* dan *outflow* uang kartal.

2.1 Analisis Deret Waktu

Time series atau deret waktu adalah hasil pengamatan yang berurutan secara kronologis terhadap variabel tertentu. Data deret waktu sering diteliti dengan harapan bisa ditemukan pola historis yang dapat dimanfaatkan dalam penyusunan ramalan atau prediksi untuk beberapa periode kedepan. Dalam mengidentifikasi pola ini, seringkali lebih mudah untuk memandang deret waktu yang terdiri dari beberapa komponen utama yaitu *trends*, siklus, variasi musiman dan komponen fluktuatif yang tidak teratur (Bowerman & O'Connell, 1993, hal. 354).

Tujuan dari analisis deret waktu secara umum terbagi menjadi dua yaitu untuk memahami atau memodelkan mekanisme proses stokastik yang menimbulkan sebuah runtun observasi dan untuk memprediksi atau meramalkan nilai dari runtun yang diamati pada periode yang akan datang berdasarkan pola data yang ada di masa lalu serta mungkin runtun waktu lain yang berhubungan atau faktor lain (Cryer & Chan, 2008, hal. 1).

2.2 Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average (ARIMA) adalah kombinasi model *Autoregressive* (AR) order p dan *Moving Average* (MA) order q dan proses *differencing* dengan order d . Secara umum model ARIMA non-musiman $ARIMA(p, d, q)$ dapat ditulis dalam persamaan (Wei, 2006, hal. 71-72):

$$\phi_p(B)(1-B)^d Y_t = \theta_0 + \theta_q(B) a_t \quad (2.1)$$

sedangkan model ARIMA musiman dapat dituliskan sebagai $ARIMA(p, d, q)(P, D, Q)^S$ dengan persamaan

$$\Phi_p(B^S)\phi_p(B)(1-B)^d Y_t = \theta_0 + \theta_q(B)\Theta_Q(B^S)a_t \quad (2.2)$$

dengan

(p, d, q)	= AR order p , <i>differencing</i> order d , MA order q untuk pola non-musiman
(P, D, Q)	= AR order P , <i>differencing</i> order D , MA order Q untuk pola musiman
S	= Periode musiman
$\phi_p(B)$	$= 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$
$\theta_q(B)$	$= 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$
$\Phi_P(B^S)$	$= 1 - \Phi_1 B^S - \Phi_2 B^{2S} - \dots - \Phi_P B^{PS}$
$\Theta_Q(B^S)$	$= 1 - \Theta_1 B^S - \Theta_2 B^{2S} - \dots - \Theta_Q B^{QS}$
a_t	= nilai residual pada waktu ke- t .

Pemodelan ARIMA meliputi 3 tahapan yaitu identifikasi, estimasi dan pengujian signifikansi parameter, dan cek diagnosa.

(i). Identifikasi

Pada tahap ini, identifikasi transformasi yang sesuai perlu dilakukan, seperti transformasi untuk menstabilkan varians dan transformasi *differencing*, keputusan untuk mengikutsertakan parameter deterministik θ_0 ketika $d \geq 1$, dan order yang tepat dari p dan q untuk model (Wei, 2006, hal. 108). Pengujian stasioneritas dalam *mean* dapat dilakukan dengan *Augmented Dickey Fuller Test*, sedangkan pengujian stasioneritas dalam varians dapat dilakukan dengan CI 95% untuk *lambda* dalam transformasi Box-Cox. Penentuan order p dan q dapat dilakukan berdasarkan pola ACF dan PACF sample.

Autocorrelation Function (ACF) merupakan korelasi antara Y_t dan Y_{t+k} dari proses yang sama pada *lag* waktu yang berbeda (Wei, 2006, hal. 10-11). ACF untuk sampel Y_1, Y_2, \dots, Y_n diperoleh melalui

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0} = \frac{\sum_{t=1}^{n-k} (Y_t - \bar{Y})(Y_{t+k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}, \quad k = 0, 1, 2, \dots \quad (2.3)$$

Partial Autocorrelation (PACF) dalam analisis *time series* adalah korelasi antara Y_t dan Y_{t+k} dengan menghilangkan korelasi antara Y_t dan Y_{t+k} dengan $Y_{t+1}, Y_{t+2}, \dots, Y_{t+k-1}$. PACF dinyatakan sebagai korelasi bersyarat $\text{Corr}(Y_t, Y_{t+k} | Y_{t+1}, Y_{t+2}, \dots, Y_{t+k-1})$ (Wei, 2006, hal. 22-23). Perhitungan PACF untuk sampel dapat diperoleh melalui

$$\hat{\phi}_{k+1,k+1} = \frac{\hat{\rho}_{k+1} - \sum_{j=1}^k \hat{\phi}_{k,j} \hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^k \hat{\phi}_{k,j} \hat{\rho}_j} \quad (2.4)$$

dan

$$\hat{\phi}_{k+1,j} = \hat{\phi}_{k,j} - \hat{\phi}_{k+1,k+1} \hat{\phi}_{k,k+1-j} \quad \text{dengan } j=1, 2, \dots, k. \quad (2.5)$$

Pendugaan model dinyatakan berdasarkan karakteristik ACF dan PACF teoritis untuk proses setasioner dengan ketentuan seperti pada Tabel 2.1 (Wei, 2006, hal. 109).

Tabel 2.1. Karakteristik ACF dan PACF secara teoritis

Proses	ACF	PACF
AR(p)	<i>Dies down</i> (turun cepat) atau turun secara eksponensial atau berpola sinusoidal	<i>Cuts off</i> setelah lag p
MA(q)	<i>Cuts off</i> setelah lag q	<i>Dies down</i> (turun cepat) atau turun secara eksponensial atau berpola sinusoidal
ARMA(p, q)	<i>Dies down</i> setelah lag $(q - p)$	<i>Dies down</i> setelah lag $(p - q)$.

(ii). Estimasi dan Pengujian Signifikansi Parameter

Estimasi parameter dari model dugaan dapat dilakukan menggunakan metode *Maximum Likelihood Estimation* (MLE). Estimasi parameter menggunakan MLE pada dasarnya terdiri dari dua tahapan yakni menentukan fungsi *likelihood* dan menentukan nilai taksiran yang memaksimumkan fungsi *likelihood* yang telah diperoleh (Hamilton, 1994, hal. 117). Misal model $AR(p)$:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + a_t. \quad (2.6)$$

dengan $a_t \sim \text{i.i.d. } N(0, \sigma^2)$. Diberikan deret waktu Y_1, Y_2, \dots, Y_n , dengan fungsi *likelihood* L merupakan *joint probability density* dari pengamatan. Sehingga untuk model $AR(p)$, L merupakan fungsi dari ϕ dan σ^2 dengan syarat Y_1, Y_2, \dots, Y_n . Dengan demikian, fungsi *likelihood* L dapat dituliskan sebagai:

$$\begin{aligned} L(a_1, a_2, \dots, a_n) &= \prod_{t=1}^n \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}(a_t)^2} \\ &= \left(\frac{1}{\sqrt{2\pi\sigma}} \right)^n e^{-\frac{1}{2}(a_1^2 + a_2^2 + \dots + a_n^2)}. \end{aligned} \quad (2.7)$$

Langkah selanjutnya untuk mempermudah perhitungan memaksimumkan fungsi *likelihood*, maka digunakan \ln untuk fungsi *likelihood* tersebut, sehingga diperoleh

$$\begin{aligned} \ln L(a_1, a_2, \dots, a_n) &= \ln \left(\frac{1}{\sqrt{2\pi\sigma}} \right)^n + \left(-\frac{1}{2\sigma^2} (a_1^2 + a_2^2 + \dots + a_n^2) \right) \\ &= \ln \left(\frac{1}{\sqrt{2\pi\sigma}} \right)^n + \left(-\frac{1}{2\sigma^2} \sum_{t=1}^n a_t^2 \right). \end{aligned} \quad (2.8)$$

Memaksimumkan fungsi \ln fungsi *likelihood* untuk parameter ϕ_1 akan sama dengan meminimumkan $\frac{1}{2\sigma^2} \left(\sum_{t=1}^n a_t^2 \right)$. Misalkan :

$$Q = \frac{1}{2\sigma^2} \left(\sum_{t=1}^n a_t^2 \right), \quad (2.9)$$

maka untuk meminimumkan Q dapat diperoleh melalui (Hamilton, 1994, hal. 118-126):

$$\frac{dQ}{d\phi} = 0. \quad (2.10)$$

Berdasarkan persamaan sebelumnya, dapat diketahui bahwa pada model AR(1) nilai a_t diperoleh dari $Y_t - \phi_1 Y_{t-1}$. Sehingga:

$$\begin{aligned} \frac{d \left[\sum_{t=1}^n (Y_t - \phi_1 Y_{t-1})^2 \right]}{d\phi_1} &= 0 \\ 2 \sum_{t=1}^n (Y_t - \phi_1 Y_{t-1})(-Y_{t-1}) &= 0 \\ \sum_{t=1}^n (-Y_t Y_{t-1} + \phi_1 Y_{t-1}^2) &= 0 \\ - \sum_{t=1}^n Y_t Y_{t-1} + \sum_{t=1}^n \phi_1 Y_{t-1}^2 &= 0. \end{aligned} \quad (2.11)$$

Berdasarkan persamaan diatas, nilai ϕ_1 dapat dituliskan sebagai:

$$\hat{\phi}_1 = \frac{\sum_{t=1}^n Y_t Y_{t-1}}{\sum_{t=1}^n \phi_1 Y_{t-1}^2}. \quad (2.12)$$

Parameter ϕ_1 dapat diketahui dengan memisalkan bahwa $a_1=0$ sehingga:

$$\hat{\phi}_1 = \frac{\sum_{t=2}^n Y_t Y_{t-1}}{\sum_{t=2}^n \phi_1 Y_{t-1}^2}. \quad (2.13)$$

Proses estimasi hingga diperoleh persamaan diatas disebut juga sebagai *conditional likelihood estimation*. Pada model AR(p), nilai $a_t = Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \dots - \phi_p Y_{t-p}$. Sehingga untuk meminimumkan persamaan (2.9), estimasi parameter untuk ϕ_1 dapat ditunjukkan dengan

$$\begin{aligned}
& \frac{\partial \left[\sum_{t=p+1}^n (Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \dots - \phi_p Y_{t-p})^2 \right]}{\partial \phi_1} = 0 \\
& 2 \sum_{t=p+1}^n (Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \dots - \phi_p Y_{t-p}) (-Y_{t-1}) = 0 \\
& \sum_{t=p+1}^n (Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \dots - \phi_p Y_{t-p}) (Y_{t-1}) = 0. \quad (2.14)
\end{aligned}$$

Persamaan (2.14) juga dapat dituliskan menjadi:

$$\begin{aligned}
\sum_{t=p+1}^n (Y_t Y_{t-1}) &= \left(\sum_{t=p+1}^n (Y_{t-1})^2 \right) \phi_1 + \left(\sum_{t=p+1}^n (Y_{t-1} Y_{t-2}) \right) \phi_2 + \dots + \\
&\left(\sum_{t=p+1}^n (Y_{t-1} Y_{t-p}) \right) \phi_p. \quad (2.15)
\end{aligned}$$

Hasil dari perhitungan $\sum_{t=p+1}^n (Y_t Y_{t-1})$ akan mendekati nilai *numerator* dari koefisien autokorelasi r_1 dengan $\bar{Y}=0$. Hal ini juga berlaku untuk $\sum_{t=p+1}^n (Y_{t-1} Y_{t-2})$ hingga $\sum_{t=p+1}^n (Y_{t-1} Y_{t-p})$. Sehingga apabila kedua sisi dibagi dengan $\sum_{t=p+1}^n (Y_t)^2$, akan diperoleh :

$$r_1 = \phi_1 + r_1 \phi_2 + r_2 \phi_3 + \dots + r_{p-1} \phi_p. \quad (2.16)$$

Estimasi parameter untuk $\phi_2, \phi_3, \dots, \phi_p$ juga dapat dilakukan dengan cara yang sama. Sehingga akan diperoleh

$$\begin{aligned}
r_1 &= \phi_1 + r_1 \phi_2 + r_2 \phi_3 + \dots + r_{p-1} \phi_p \\
r_2 &= r_1 \phi_1 + \phi_2 + r_1 \phi_3 + \dots + r_{p-2} \phi_p \\
&\vdots \\
r_p &= r_{p-1} \phi_1 + r_{p-2} \phi_2 + r_{p-3} \phi_3 + \dots + \phi_p. \quad (2.17)
\end{aligned}$$

Persamaan (2.17) disebut juga persamaan Yule-Walker untuk model $AR(p)$ (Cryer & Chan, 2008, hal. 72). Berdasarkan persamaan (2.17), maka nilai $\phi_1, \phi_2, \phi_3, \dots, \phi_p$ dapat ditentukan melalui

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} 1 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & 1 & r_1 & \cdots & r_{p-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & 1 \end{bmatrix}^{-1} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_p \end{bmatrix}. \quad (2.18)$$

Setelah diperoleh nilai estimasi dan *standard error* dari parameter-parameter model ARIMA, tahapan selanjutnya adalah melakukan pengujian signifikansi parameter. Misalkan untuk uji signifikansi parameter model MA(q) hipotesis yang digunakan adalah

$$H_0: \theta_j = 0$$

$$H_1: \theta_j \neq 0, \text{ dengan } j=1,2,\dots,q$$

Statistik uji yang digunakan adalah

$$t_{\text{hitung}} = \frac{\hat{\theta}_j}{SE(\hat{\theta}_j)}. \quad (2.19)$$

H_0 ditolak apabila nilai statistik uji $|t_{\text{hitung}}| > t_{\alpha/2, (n-n_p)}$ dengan n merupakan banyaknya pengamatan dan n_p merupakan banyaknya parameter yang diestimasi.

(iii). Cek Diagnosa

Model ARIMA harus memenuhi asumsi *white noise* (residual bersifat identik dan independen) serta berdistribusi normal. Pengujian asumsi independen dapat dilakukan menggunakan Ljung-Box *test*. Pengujian ini dilakukan menggunakan autokorelasi dari residual sampel. Hipotesis yang digunakan untuk uji Ljung-Box adalah (Wei, 2006, hal. 152-153)

$$H_0 : \rho_1 = \rho_2 = \dots = \rho_k = \dots = \rho_K \text{ (residual independen),}$$

$$H_1 : \text{minimal ada satu nilai } \rho_k \neq 0 \text{ dengan } k = 1, 2, \dots, K \text{ (residual tidak independen).}$$

perhitungan statistik uji Q dapat dilakukan menggunakan

$$Q = n(n+2) \sum_{k=1}^K \frac{\hat{\rho}_k^2}{n-1} \quad (2.20)$$

H_0 ditolak apabila nilai $Q > \chi_{\alpha, K-p-q}^2$ dengan $\hat{\rho}_k$ merupakan autokorelasi dari residual, nilai p adalah banyaknya parameter AR pada model, q adalah banyaknya parameter MA pada model, n adalah banyaknya pengamatan, dan α adalah taraf signifikansi yang digunakan.

Pengujian residual bersifat identik (homogen) dapat dilakukan menggunakan pengujian efek ARCH yang diperkenalkan oleh Engle (Wei, 2006, hal. 369-370). Model AR(s) untuk \hat{n}_{t-s}^2 dapat ditunjukkan oleh

$$\hat{n}_t^2 = \theta_0 + \theta_1 \hat{n}_{t-1}^2 + \dots + \theta_s \hat{n}_{t-s}^2 + a_t, \quad (2.21)$$

untuk $t = s+1, s+2, \dots, n$. Dengan hipotesis yang digunakan adalah:

$$H_0: \theta_1 = \dots = \theta_s = 0$$

$$H_1: \text{minimal terdapat terdapat satu } \theta_j \neq 0, \text{ untuk } j=1, 2, \dots, s.$$

Apabila H_0 gagal ditolak maka estimasi untuk θ_j mendekati nol dan mengakibatkan koefisien determinasi (R^2) bernilai kecil. Statistik uji yang digunakan ditunjukkan oleh

$$LM = (n^* - s)R^2 \quad (2.22)$$

dengan $R^2 = 1 - \frac{SSE}{SST}$ dari model regresi pada persamaan tersebut

dan n^* menyatakan banyaknya pengamatan yang dipakai. Apabila nilai LM lebih besar daripada nilai $\chi^2(s)$, maka terdapat efek ARCH pada residual, dengan kata lain, residual tidak memenuhi asumsi identik.

Pengujian distribusi normal untuk residual dapat dilakukan menggunakan uji Kolmogorov-Smirnov. Berikut adalah hipotesis yang digunakan (Daniel W. , 1989):

$$H_0: F(a_i) = F_0(a_i) \text{ (Residual mengikuti distribusi normal),}$$

$$H_1: F(a_i) \neq F_0(a_i) \text{ (Residual tidak mengikuti distribusi normal).}$$

Statistik uji yang digunakan adalah

$$D = \text{Sup}\{D^+, D^-\} \quad (2.23)$$

dimana

$$D^+ = \text{Sup}|F(a_t) - F_0(a_t)| \quad (2.24)$$

$$D^- = \text{Sup}|F_0(a_t) - F(a_{t-1})| \quad (2.25)$$

dengan

$F(a_t)$ = fungsi distribusi frekuensi kumulatif residual,

$F_0(a_t)$ = fungsi distribusi frekuensi kumulatif distribusi normal,

Sup = nilai maksimum.

H_0 ditolak apabila nilai D lebih besar dari nilai tabel Kolmogorov-Smirnov yaitu $d_{n,\alpha}$ dengan n adalah banyaknya pengamatan dan α adalah taraf signifikansi yang digunakan.

2.3 ARIMAX

Secara umum, regresi *time series* sama seperti regresi linier pada umumnya, namun data *time series* dalam bidang ekonomi sering disusun dalam periode bulanan, kuartal atau tahunan dan memiliki pola musiman serta efek variasi kalender (Ahmad, Setiawan, Suhartono, & Masun, 2015). Contoh model regresi *time series* yang memuat pola musiman, *trend*, dan variasi kalender adalah sebagai berikut

$$Y_t = \beta_1 L_{1,t} + \beta_2 L_{2,t} + \dots + \beta_w L_{w,t} + \phi t + \gamma_1 M_{1,t} + \gamma_2 M_{2,t} + \dots + \gamma_s M_{s,t} + N_t \quad (2.26)$$

Asumsi yang harus dipenuhi dari model tersebut adalah N_t harus *white noise* (Ahmad, Setiawan, Suhartono, & Masun, 2015). Apabila asumsi tersebut belum dipenuhi maka dilakukan pemodelan ARIMA terhadap residual N_t (Saputri, Suhartono, & Prastya, 2017). Sehingga model umum ARIMAX dengan efek variasi kalender adalah dinyatakan sebagai

$$Y_t = \beta_1 L_{1,t} + \dots + \beta_w L_{w,t} + \phi t + \gamma_1 M_{1,t} + \dots + \gamma_s M_{s,t} + \frac{\theta_q(B)}{\phi_p(B)} a_t \quad (2.27)$$

dengan

$L_{1,t}, L_{2,t}, \dots, L_{w,t}$ = variabel *dummy* untuk efek variasi

	kalender
$M_{1,t}, M_{2,t}, \dots, M_{s,t}$	= variabel <i>dummy</i> untuk bulan
t	= variabel trend
$\phi_p(B)$	$= 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$
$\theta_q(B)$	$= 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$

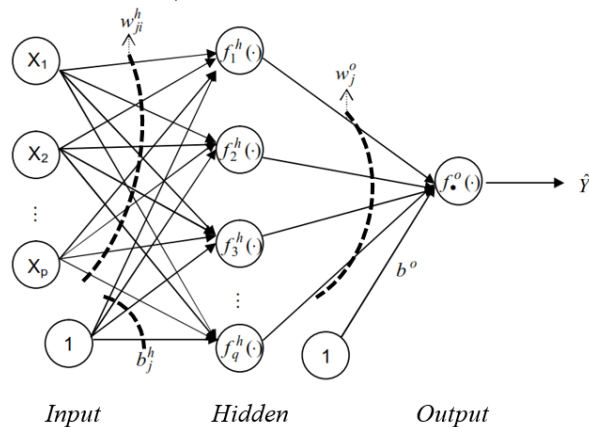
2.4 Deep Learning for Time Series

Artificial Neural Networks (ANN) adalah pendekatan komputasi yang berdasar pada kumpulan neuron untuk memproses informasi menggunakan teori koneksi. Pendekatan *machine learning* ini mensimulasikan fungsi pengenalan pola dari jaringan syaraf dalam otak manusia. Jaringan bio-elektrik dalam otak manusia terdiri dari banyak neuron dan synapsis dan fungsi penghubung untuk menghubungkan neuron (Thinsungneon, Kerdprasop, & Kerdprasop, 2017). AI memiliki cakupan yang sangat luas salah satu yang terkenal dalam analisis data adalah bidang *machine learning*. Pada awal tahun 2006, *deep learning* mulai diperkenalkan secara luas sebagai bagian dari *machine learning* dan banyak digunakan untuk klasifikasi, *forecasting* dan lain lain (Patterson & Gibson, 2017, hal. 6).

Deep learning dalam peramalan memiliki berbagai macam jenis dan sebutan yang berbeda dalam beberapa sumber referensi. *Deep feedforward networks* juga sering disebut *feedforward neural networks* atau *multi-layers perceptrons* (MLPs) merupakan model *deep learning* klasik. Tujuan dari sebuah *feedforward network* adalah mengestimasi beberapa fungsi *aktivasi* (Goodfellow, Bengio, & Courville, 2016, hal. 167). Dalam peramalan, metode *neural network* yang banyak digunakan adalah *feedforward neural network* (FFNN). Pada FFNN, proses dimulai dari *input* yang diterima oleh *neuron*, dimana *neuron-neuron* ini dikelompokkan dalam *layer*. Informasi yang diterima dari *layer input* dilanjutkan ke *layer-layer* dalam FFNN secara berurutan hingga mencapai *layer output*. *Layer* yang berbeda diantara *input* dan *output* disebut

hidden layer (Suhartono, 2007, hal. 44). *Input* yang digunakan dalam FFNN untuk peramalan adalah *lag* dari observasi sebelumnya dengan *output* merupakan hasil peramalan. Pemilihan *input* dilakukan berdasarkan *lag* yang signifikan pada plot PACF data yang telah stasioner (Crone & Kourentzes, 2009). FFNN dapat diterapkan pada permasalahan pola non-linier. Selanjutnya dalam penelitian ini, FFNN dengan lebih dari 1 *hidden layer* ini akan disebut sebagai *deep neural network* (DNN) sehingga jika dalam penelitian ini disebutkan DNN maka yang dimaksud adalah *deep learning feedforward neural network* dengan 2 *hidden layer*.

Dalam pemodelan statistik, FFNN dapat dipandang sebagai suatu kelas yang fleksibel dari fungsi-fungsi non-linier. Secara umum, model ini bekerja dengan menerima suatu vektor dari input \mathbf{x} dan kemudian menghitung suatu respon atau output $\hat{y}(\mathbf{x})$ dengan memproses (*propagating*) \mathbf{x} melalui elemen-elemen proses yang saling terkait. FFNN dengan satu *hidden layer* yang terdiri dari q unit neuron dan *output layer* yang terdiri dari satu unit neuron digambarkan dalam arsitektur seperti Gambar 2.1 (Suhartono, 2007, hal. 45).



Gambar 2.1. Arsitektur FFNN 1 *hidden layer*

Dalam arsitektur Gambar 2.1 nilai nilai respon atau output \hat{y} dapat dihitung dengan

$$\hat{y}_{(k)} = f^0 \left[\sum_{j=1}^q \left[w_j^0 f_j^h \left(\sum_{i=1}^p w_{ji}^h x_{i(k)} + b_j^h \right) + b^0 \right] \right] \quad (2.28)$$

dengan

$x_{i(k)}$: variabel input sebanyak p , ($i = 1, 2, \dots, p$)

$\hat{y}_{(k)}$: nilai dugaan dari variabel output

k : indeks pasangan data input-target $(x_{i(k)}, \hat{y}_{(k)})$, $k = 1, 2, \dots, n$

w_{ji}^h : bobot dari input ke- i yang menuju neuron ke- j pada *hidden layer*, ($j = 1, 2, \dots, q$)

b_j^h : bias pada neuron ke- j pada *hidden layer* ($j = 1, 2, \dots, q$)

f_j^h : fungsi aktivasi di neuron ke- j pada *hidden layer*

w_j^0 : bobot dari neuron ke- j di *hidden layer* yang menuju neuron pada *output layer*

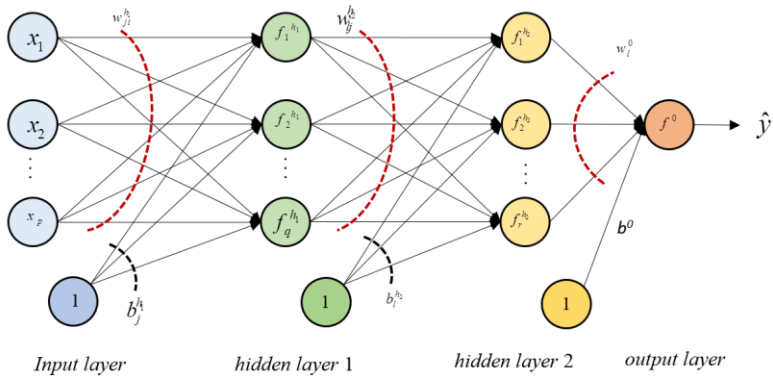
b^0 : bias pada neuron di *output layer*

f^0 : fungsi aktivasi pada neuron di *output layer*.

Pemetaan secara keseluruhan yang terjadi pada FFNN ini selanjutnya dapat dituliskan dalam bentuk

$$\begin{bmatrix} \hat{y}_{(1)} \\ \hat{y}_{(2)} \\ \vdots \\ \hat{y}_{(n)} \end{bmatrix} = \begin{bmatrix} F(x_{1(1)}, x_{2(1)}, \dots, x_{p(1)}) \\ F(x_{1(2)}, x_{2(2)}, \dots, x_{p(2)}) \\ \vdots \\ F(x_{1(n)}, x_{2(n)}, \dots, x_{p(n)}) \end{bmatrix} \quad (2.29)$$

Pemilihan model *neural network* terbaik dapat dilakukan menggunakan *cross validation*. Metode *cross validation* merupakan metode yang banyak digunakan dalam pemilihan model *neural network* (Anders & Korn, 1999). Berdasarkan arsitektur model FFNN tersebut, maka dapat digeneralisir untuk dua *hidden layer* sehingga arsitektur DNN yang terbentuk adalah seperti pada Gambar 2.2.



Gambar 2.2 *Artsitektur Deep Neural Networks*

Sedangkan perhitungan nilai-nilai respon atau output \hat{y} DNN dapat digeneralisir dari FFNN satu layer sehingga menjadi

$$\hat{y}_{(k)} = f^0 \left[\sum_{l=1}^r \left[w_l^0 f_l^{h_2} \left(\sum_{j=1}^q w_{ji}^{h_2} f_j^{h_1} \left(\sum_{i=1}^p w_{ji}^{h_1} x_{i(k)} + b_j^{h_1} \right) + b_l^{h_2} \right) + b^0 \right] \right] \quad (2.30)$$

dengan

$x_{i(k)}$: variabel input sebanyak p , ($i = 1, 2, \dots, p$)

$\hat{y}_{(k)}$: nilai dugaan dari variabel output

k : indeks pasangan data input-target ($x_{i(k)}, \hat{y}_{(k)}$), $k = 1, 2, \dots, n$

$w_{ji}^{h_1}$: bobot dari input ke- i yang menuju neuron ke- j pada *hidden layer 1*, ($j = 1, 2, \dots, q$)

$b_j^{h_1}$: bias pada neuron ke- j pada *hidden layer 1* ($j = 1, 2, \dots, q$)

$f_j^{h_1}$: fungsi aktivasi di neuron ke- j pada *hidden layer 1*

w_l^0 : bobot dari neuron ke- l di *hidden layer 2* yang menuju neuron pada *output layer*

b^0 : bias pada neuron di *output layer*

f^0 : fungsi aktivasi pada neuron di *output layer*

- w_{ij}^h : bobot dari neuron ke- j pada *hidden layer* 1 yang menuju neuron ke- l pada *hidden layer* 2, ($j = 1, 2, \dots, q$; $l = 1, 2, \dots, r$)
- $b_l^{h_2}$: bias pada neuron ke- l pada *hidden layer* 2 ($l = 1, 2, \dots, r$)
- $f_l^{h_2}$: fungsi aktivasi di neuron ke- l pada *hidden layer* 2.

Dalam penerapan *feedforward neural network* saat ini, jenis fungsi aktivasi utama yang banyak digunakan yaitu fungsi logit sigmoid yang ditulis sebagai $(1+e^{-x})^{-1}$. Fungsi logit sigmoid memiliki range dari 0 sampai 1 (Lasheras, Vilan, Nieto, & Diaz, 2010). Fungsi logit sigmoid digunakan karena menurut Makridakis dkk (2018) fungsi logit sigmoid merupakan fungsi standar untuk *neural network* yang dinilai cukup sederhana.

2.5 Singular Spectrum Analysis

Singular spectrum analysis (SSA) adalah teknik analisis deret waktu dan peramalan yang menggabungkan unsur analisis deret waktu klasik, statistika multivariat, geometri multivariat, *dinamical systems* dan *signal processing*. SSA bertujuan mendekomposisi data deret waktu menjadi sejumlah komponen yang dapat diinterpretasikan sebagai *trend* yang perlahan bervariasi, komponen *oscillatory* (musiman) dan *noise* yang tidak terstruktur (*random*). SSA menggunakan *singular value decomposition* (SVD) dari matriks tertentu yang dibangun dari data deret waktu. SSA memiliki cakupan penerapan yang sangat luas karena tidak memerlukan asumsi-asumsi khusus seperti stasioneritas, model parameterik dan lain lain (Golyandina & Zhigljavsky, 2013, hal. 1).

Secara garis besar, SSA memiliki 2 tahap utama yaitu tahap dekomposisi dan tahap rekonstruksi. Tahap dekomposisi terdiri dari 2 langkah yaitu *embedding* dan *SVD* sedangkan tahap rekonstruksi terdiri dari 2 langkah yaitu *eigentriple grouping* dan *diagonal averaging* sehingga secara keseluruhan SSA terdiri dari 4 langkah sebagai berikut (Golyandina & Zhigljavsky, 2013, hal. 11).

2.5.1 Embedding

Diberikan data deret waktu sebagai $Y = (y_1, \dots, y_n)$ dengan panjang n dengan $n > 2$ dan Y adalah deret waktu bukan nol sehingga minimal ada 1 $y_i \neq 0$. Tahap *embedding* dilakukan dengan memetakan data deret waktu berukuran n asli kedalam *lagged vectors* berukuran L sehingga akan terbentuk sebanyak K *lagged vectors*, dengan L yang disebut *window length* berupa bilangan integer ($1 < L < n$) dan $K = n - L + 1$.

$$X_i = (y_1, \dots, y_{i+L-1})^T; (1 \leq i \leq K) \quad (2.31)$$

vektor X_i disebut sebagai *L-lagged vectors* sedangkan *trajectory matrix* (matriks lintasan) \mathbf{X} dari data deret waktu F diberikan sebagai berikut

$$\mathbf{X} = [X_1 : \dots : X_K] = (x_{ij})_{i,j=1}^{L,K} = \begin{pmatrix} y_1 & y_2 & y_3 & \dots & y_K \\ y_2 & y_3 & y_4 & \dots & y_{K+1} \\ y_3 & y_4 & y_5 & \dots & y_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & y_{L+2} & \dots & y_n \end{pmatrix} \quad (2.32)$$

lagged vectors X_i merupakan kolom dari matriks lintasan \mathbf{X} . Baris dan kolom dari \mathbf{X} adalah subderet dari data deret waktu asli. Unsur ke- (i, j) dari matriks \mathbf{X} adalah $x_{ij} = y_{i+j-1}$. Matriks lintasan \mathbf{X} merupakan sebuah matriks Hankel. Jika n dan L ditetapkan, maka ada korespondensi satu-satu antara matriks lintasan dan deret waktu (Golyandina & Zhigljavsky, 2013, hal. 12-13).

Penentuan *Window Length* merupakan salah satu tahapan yang penting. Jika nilai *window length* yang ditentukan tidak sesuai, *source signals* tidak dapat diekstrak dari *mixed signal*. Hasil teoritis mengkonfirmasi bahwa untuk sebuah deret waktu yang panjang (nilai n besar), nilai yang cocok untuk *windows length* adalah median dari panjang deret waktu tersebut (Golyandina & Zhigljavsky, 2013, hal. 47-48). Penentuan besar *window length* L yang digunakan dalam SSA tidak memiliki aturan yang baku. Terdapat beberapa aturan yang terangkum berdasarkan Golyandina,

Nekrutkin, & Zhigljavsky (2001) mengenai penentuan nilai *window length*

- Pada SVD akan ditunjukkan matriks dengan *window length* L adalah ekuivalen dengan matriks dari komplementer *window length* yaitu $K = n - L + 1$. Namun, penambahan nilai L diatas setengah dari jumlah data akan menghasilkan seperti hasil yang telah dilakukan dengan nilai L yang lebih kecil
- Semakin besar nilai L maka dekomposisi yang dihasilkan akan semakin detail. Sehingga nilai L yang digunakan lebih baik dengan nilai besar namun lebih kecil dari jumlah setengah data
- Jika diketahui periodik dari data maka nilai L yang lebih baik adalah mengikuti pola periodik data tersebut (Zhang, Wang, He, & Peng, 2011).

2.5.2 Singular Value Decomposition (SVD)

SVD dilakukan pada matriks lintasan \mathbf{X} yang telah diperoleh. Diberikan matriks $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ dan $\lambda_1, \dots, \lambda_L$ merupakan *eigenvalues* dari \mathbf{S} dengan $\lambda_1 \geq \dots \geq \lambda_L \geq 0$ dan U_1, \dots, U_L adalah sistem ortonormal dari eigen vektor matriks \mathbf{S} yang bersesuaian dengan *eigenvalues*. Diberikan $d = \text{rank } \mathbf{X} = \max\{i, \text{dengan } \lambda_i > 0\}$ dan $V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i}, (i = 1, \dots, d)$ sehingga SVD dari matriks lintasan \mathbf{X} dapat ditulis sebagai

$$\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d, \quad (2.33)$$

dimana $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$. Rank matriks \mathbf{X}_i adalah 1 sehingga disebut *elementary matrix*. Kumpulan $(\sqrt{\lambda_i}, U_i, V_i)$ disebut *eigentriple* ke- i dari SVD (Golyandina & Zhigljavsky, 2013, hal. 14).

2.5.3 Eigentriple grouping

Setelah ekspansi dari SVD diperoleh, prosedur pengelompokan akan mempartisi himpunan indeks $\{1, \dots, d\}$ kedalam m himpunan bagian yang saling lepas I_1, \dots, I_m . Jika $I = \{i_1, \dots, i_p\}$ maka matriks \mathbf{X}_I yang dihasilkan yang bersesuaian dengan I dijelaskan sebagai $\mathbf{X}_I = \mathbf{X}_{i_1} + \dots + \mathbf{X}_{i_p}$. Matriks ini dihitung untuk nilai $I = I_1, \dots, I_m$ dan ekspansi SVD sebelumnya menyebabkan dekomposisi

$$\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_d} \quad (2.34)$$

Prosedur pemilihan himpunan I_1, \dots, I_m disebut sebagai pengelompokan *eigentriple*. Jika $m = d$ dan $I_j = \{j\}$, serta $j = 1, \dots, d$, maka pengelompokan yang sesuai disebut elementer (Golyandina & Zhigljavsky, 2013, hal. 16-18).

2.5.4 Diagonal Averaging

Tahap ini mentransformasi \mathbf{X}_{I_j} dari dekomposisi yang sudah terkelompok kedalam deret waktu baru dengan panjang n . Diberikan \mathbf{Z} adalah matriks berukuran $L \times K$ dengan elemen z_{ij} , $1 \leq i \leq L$, $1 \leq j \leq K$ dan $L^* = \min(L, K)$, $K^* = \max(L, K)$, $n = L + K - 1$. Jika $L < K$ maka $z_{ij}^* = z_{ij}$ sedangkan jika $L \geq K$ maka $z_{ij}^* = z_{ji}$. Dengan melakukan *diagonal averaging* kita mentransfer matriks \mathbf{Z} kedalam deret waktu z_1, \dots, z_n menggunakan formula

$$z_k \left\{ \begin{array}{ll} \frac{1}{k} \sum_{m=1}^k z_{m, k-m+1}^* & \text{untuk } 1 \leq k < L^* \\ \frac{1}{L} \sum_{m=1}^{L^*} z_{m, k-m+1}^* & \text{untuk } L^* \leq k \leq K^* \\ \frac{1}{n-k} \sum_{m=k-K^*+1}^{n-K^*+1} z_{m, k-m+1}^* & \text{untuk } K^* < k \leq n \end{array} \right. \quad (2.35)$$

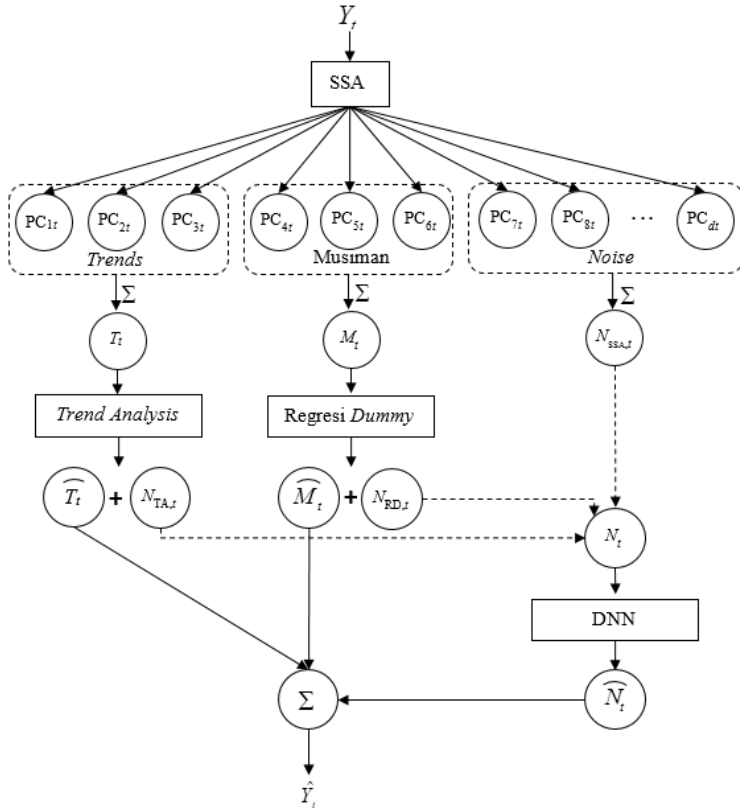
Hal ini sesuai dengan rata-rata elemen matriks diatas ‘*antidiagonals*’ $i+j=k+1$: pilihan $k=1$ memberikan $z_1 = zy_{1,1}$ untuk $k=2$ maka $z_2 = (z_{1,2} + z_{2,1})/2$ dan seterusnya. *Diagonal averaging* dilakukan pada matriks \mathbf{X}_{I_k} untuk membuat deret waktu yang tersusun $\mathbf{Z}^{(k)} = (z_1^{(k)}, \dots, z_n^{(k)})$ sehingga deret waktu awal y_1, \dots, y_n telah terdekomposisi kedalam sejumlah m deret waktu yang terekonstruksi:

$$z_u = \sum_{k=1}^m z_u^{(k)}; (u = 1, 2, \dots, n) \quad (2.36)$$

Deret waktu yang terekonstruksi terbentuk dari pengelompokan elementer yang akan disebut *elementary reconstructed series* (Golyandina & Zhigljavsky, 2013, hal. 18-19). Dalam penelitian ini, deret waktu yang terekonstruksi dari setiap matriks dekomposisi sesuai persamaan (2.38) selanjutnya disebut sebagai *principal component* (PC).

2.6 Model Hybrid Singular Spectrum Analysis & Deep Neural Network (SSA-DNN)

Secara umum, metode SSA mampu mendekomposisi suatu deret data menjadi pola tren, musiman dan *noise*. Dari hasil dekomposisi pola-pola data tersebut kemudian yang akan dilakukan peramalan dengan menggunakan beberapa metode peramalan salah satunya *deep neural network* (DNN). Peramalan yang dilakukan dapat menggunakan peramalan secara individu maupun agregat. Peramalan individu dilakukan dengan meramalkan setiap komponen utama yang terbentuk tanpa menggabungkan sebagai komponen tren, musiman atau *noise*. Sedangkan peramalan secara agregat yaitu dilakukan dengan menjumlahkan komponen yang memiliki pola yang sama sehingga hanya terbentuk 3 pola utama yaitu tren, musiman dan *noise*. Apabila *noise* yang dihasilkan telah *white noise* maka tidak perlu untuk dilakukan peramalan. Hasil dari peramalan baik pada peramalan secara individu ataupun agregat akan dilakukan penjumlahan karena jenis data yang digunakan adalah dekomposisi sehingga akan terbentuk nilai ramalan (Setyowati & Suhartono, 2017). Zubaidi (2018) pernah menggunakan SSA sebagai tahap *preprocessing* ANN dalam meramalkan konsumsi air bulanan. Model *hybrid* SSA-DNN ini akan menggunakan hasil dekomposisi dari SSA yang polanya tidak teridentifikasi (*noise*) sebagai variabel input *deep learning* (DNN). Metode *hybrid* SSA-DNN yang dilakukan dalam penelitian ini secara umum diilustrasikan seperti pada Gambar 2.3.



Gambar 2.3 Ilustrasi Metode Hybrid SSA-DNN

Gambar 2.3 menunjukkan hasil dekomposisi SSA dimodelkan menggunakan *trend analysis* untuk PC yang diidentifikasi sebagai tren, regresi *dummy* untuk PC yang diidentifikasi sebagai musiman dan DNN untuk PC yang tidak teridentifikasi (*noise*). Gambar 2.3 menggambarkan pemodelan secara agregat dengan menjumlahkan PC yang diidentifikasi sebagai pola tren menjadi T_t dan menjumlahkan PC yang diidentifikasi sebagai pola musiman menjadi M_t . Sedangkan PC yang tidak teridentifikasi dijumlahkan dengan residual *trend analysis* dan regresi *dummy* untuk dimodelkan menggunakan DNN.

2.7 Kriteria Evaluasi Model

Pemilihan model terbaik dilakukan jika terdapat lebih dari satu model yang dapat digunakan. Dalam pendekatan *out-sample* pemilihan model terbaik dilakukan berdasarkan *error* dari hasil ramalan. Kriteria yang digunakan adalah *Root Mean Square Error Prediction* (RMSEP) dan *Symmetric Mean Absolute Percentage Error* (SMAPE). RMSE dan SMAPE dapat dihitung dengan rumus sebagai berikut (Ahmad, Setiawan, Suhartono, & Masun, 2015).

$$RMSEP = \sqrt{\frac{1}{L} \sum_{l=1}^L (Y_{n+l} - \hat{Y}_n(l))^2} \quad (2.37)$$

$$sMAPE = \left(\frac{1}{L} \sum_{l=1}^L \frac{2|Y_{n+l} - \hat{Y}_n(l)|}{(|Y_{n+l}| + |\hat{Y}_n(l)|)} \right) \times 100 \quad (2.38)$$

dengan:

- Y_{n+l} : data *out-sample* yang ke- l ($l = 1, 2, \dots, L$)
- $\hat{Y}_n(l)$: ramalan untuk data *out-sample* yang ke- l
- L : panjang data *out-sample*

2.8 Penelitian Sebelumnya

Pada beberapa tahun terakhir, telah banyak dikembangkan penelitian yang membahas tentang *singular spectrum analysis*, *deep neural network* beserta metode yang termasuk kedalam *machine learning* untuk dikombinasikan bersama berbagai metode lain untuk dibandingkan. Kesimpulan yang dihasilkan pun juga cukup beragam ketika membandingkan model yang secara matematis terbilang rumit terhadap model yang secara matematis relatif lebih sederhana untuk dituliskan. Beberapa penelitian dengan metode yang berhubungan dengan penelitian ini dirangkum dalam Tabel 2.2

Tabel 2.2 Penelitian Sebelumnya

No	Penulis	Studi Kasus	Model	Model Terbaik
1	Makridakis, S (2000)	3003 data <i>time series</i> M3	Naive/simple, Explicit trend, Dekomposisi, ARIMA/ARARMA, Expert systems, NN	Bervariasi: Model lebih rumit tidak selalu lebih akurat dibanding model sederhana
2	Bai, Y et.al (2016)	<i>Daily Reservoir Inflow</i>	Multiscale Deep Feature Learning) MDL, D-NN, BPNN, LSSVR, WANN	MDL
3	Saputri, P. D (2017)	Inflow & Outflow Indonesia	QRNN, NN, ARIMAX	QRNN
4	Alzahrani, A et.al (2017)	Solar Irradiance	FNN, SVR, DRNN with LSTM	DRNN
5	He, W (2017)	<i>Power Load (Short Term Forecasting)</i>	Regresi Linier, DNN, SVR, CNN-RNN, Parallel CNN-RNN	DNN
6	Setyowati, E. (2018)	Inflow & Outflow Indonesia	ARIMAX, SSA-NN	SSA-NN
7	Makridakis, S (2018)	100.000 data <i>time series</i> M4	Model hybrid, model Kombinasi, model statistik, model machine learning	Hybrid
8	Makridakis, S et.al (2018)	1045 data <i>time series</i>	model statistik (ETS, ARIMA, Damped, Theta, SES, Naive 2, Holt), model ML (MLP, BNN, RBF, GRNN, KNN, CART, SVR, GP, RNN, LSTM)	Model statistik
9	Liu, H et.al (2018)	Kecepatan Angin	ARIMA, LSTM, ELM, VMD-ELM, VMD-LSTM-ELM, EMD-SSA-LSTM-ELM, WPD-LSTM-ELM, VMD-SSA-LSTM-ELM	VMD-SSA-LSTM-ELM
10	Lago, J et.al (2018)	<i>Electricity Price</i>	DNN, Hybrid DNN, AR, GARCH Based Model, ARX, Regime switching, Semiparametrik, AI (SVR, NN dll), Ensemble (Random forest dll)	DNN > Hybrid DNN > Lainnya

2.9 White Test

White test dalam penelitian ini digunakan untuk menguji apakah terdapat hubungan nonlinier antara suatu deret waktu dengan lag-lagnya. Pengujian ini dikembangkan dari model *neural network* yang ditemukan White pada 1989. *White test* menguji hubungan non-linier antara input dan output dengan 1 *hidden layer* (Lee, White & Granger, 1993). Model yang digunakan pada uji *White* dinyatakan dalam persamaan

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p} + \theta_1 X_{1,t}^* + \theta_2 X_{2,t}^* + a_t \quad (2.39)$$

dengan $X_{1,t}^*$ dan $X_{2,t}^*$ merupakan komponen non-linier pada uji white yang diturunkan dari model NN. Hipotesis yang digunakan adalah sebagai berikut

H_0 : $\theta_1 = \theta_2 = 0$ (Tidak ada hubungan non-linier)

H_1 : minimal ada 1 $\theta_i \neq 0$, dengan $i=1, 2$ (Terdapat hubungan non-linier)

Statistik uji yang digunakan adalah F yang dihitung melalui langkah-langkah sebagai berikut

1. Meregresikan Y_t dengan sejumlah p variabel prediktor seperti dalam persamaan (2.39) tanpa mengikutsertakan komponen non-linier, sehingga diperoleh jumlah kuadrat residualnya yang dinamai sebagai SSR_0
2. Meregresikan residual pada langkah 1 dengan semua variabel prediktor dalam persamaan (2.39) sehingga diperoleh jumlah kuadrat residualnya yang dinamai sebagai SSR_1
3. Menghitung statistik uji F melalui

$$F_{hitung} = \frac{(SSR_0 - SSR_1) / 2}{SSR_1 / (n - p - 1 - 2)} \quad (2.40)$$

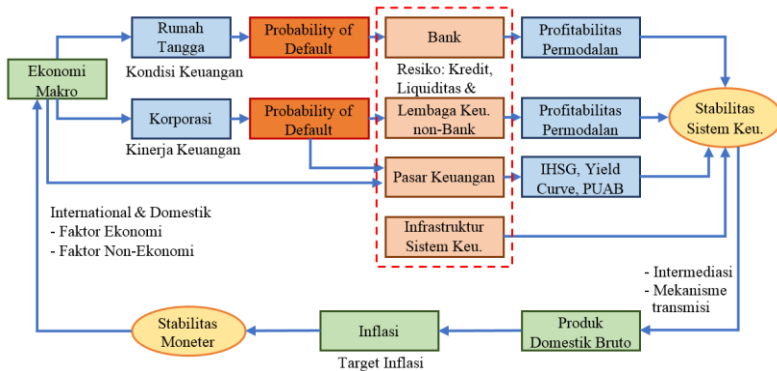
dengan n merupakan banyaknya pengamatan yang digunakan. Statistik uji yang digunakan mengikuti distribusi F dengan derajat bebas $v_1=2$ dan $v_2=(n - p - 1 - 2)$.

2.10 *Inflow dan Outflow Uang Kartal*

Keberadaan uang kartal sangat penting bagi masyarakat dan perekonomian sebuah negara karena uang kartal secara fungsi uang kartal dipakai sebagai alat pembayaran yang sah menggantikan sistem barter, sebagai alat penyimpan nilai dan satuan penghitung. Uang kartal disirkulasikan ke seluruh wilayah negara sesuai dengan kebutuhan gerak roda perekonomiannya. Bila kebutuhan uang yang beredar tidak tercukupi maka perekonomian negara akan terganggu dan tidak stabil bahkan kolaps. Sedangkan jika jumlah uang melebihi kebutuhan, nilai mata uang akan turun sehingga keseimbangan antara suplai dan permintaan perlu dijaga agar sesuai kebutuhan termasuk menjelang hari raya seperti Idul Fitri. Bank Indonesia membuat Rencana Kebutuhan Uang (RKU) untuk memenuhi kebutuhan uang di masyarakat dalam jumlah nominal yang cukup, jenis pecahan yang sesuai, tepat waktu dan dalam kondisi layak edar, mengingat Bank Indonesia merupakan satu-satunya institusi yang berhak mengeluarkan, mengedarkan, menarik dan memusnahkan uang menurut Undang-Undang. RKU mempertimbangkan berbagai aspek salah satunya adalah *inflow* dan *outflow* menurut data historis yang ada (Bank Indonesia, 2011).

Transaksi penarikan uang rupiah (*outflow*) merupakan informasi mengenai aliran uang kertas dan uang logam yang keluar dari Bank Indonesia kepada perbankan dan masyarakat, terdiri dari penarikan bank umum, penarikan non-bank, kas keliling dalam rangka penukaran, penarikan dalam rangka kas titipan di bank umum, dan penarikan lainnya. Sedangkan transaksi penyetoran uang rupiah (*inflow*) merupakan informasi mengenai aliran uang kertas dan uang logam yang masuk dari perbankan dan masyarakat ke Bank Indonesia, terdiri dari setoran bank umum, setoran non-bank, kas keliling dalam rangka hasil penukaran, penyetoran dalam rangka kas titipan di bank umum, dan penyetoran lainnya (Bank Indonesia, 2018). *Inflow* dan *outflow* tentu akan sangat mempengaruhi jumlah uang yang beredar sehingga menjadi kunci utama dalam keseimbangan jumlah uang beredar. Keseimbangan jumlah uang yang beredar di masyarakat berhubungan erat dengan

stabilitas keuangan dan stabilitas moneter. Hubungan tersebut secara umum dapat diilustrasikan dalam Gambar 2.5.



Gambar 2.4 Ilustrasi Hubungan Stabilitas Sistem Keuangan dan Stabilitas Moneter

Inflow dan *outflow* punya andil dalam menstabilkan jumlah uang beredar. Uang Beredar adalah kewajiban sistem moneter (Bank Sentral, Bank Umum, dan Bank Perkreditan Rakyat/BPR) terhadap sektor swasta domestik (tidak termasuk pemerintah pusat dan bukan penduduk). Kewajiban yang menjadi komponen Uang Beredar terdiri dari uang kartal yang dipegang masyarakat (di luar Bank Umum dan BPR), uang giral, uang kuasi yang dimiliki oleh sektor swasta domestik, dan surat berharga selain saham yang diterbitkan oleh sistem moneter yang dimiliki sektor swasta domestik dengan sisa jangka waktu sampai dengan satu tahun.

BAB III METODOLOGI PENELITIAN

Metodologi penelitian akan menjelaskan tentang kajian simulasi dan kajian terapan yang mencakup tentang data yang digunakan, variabel penelitian, langkah analisis beserta diagram alirnya.

3.1 Kajian simulasi

Kajian simulasi bertujuan untuk mengetahui performa metode ARIMAX, DNN dan *hybrid* SSA-DNN dalam menangkap pola *trend*, musiman, efek variasi kalender dan *noise* pada data simulasi. Data simulasi dibangkitkan dengan memberikan 2 skenario berbeda. Secara umum data simulasi dapat disusun dalam persamaan

$$Y_t = T_t + M_t + L_t + N_t \quad (3.1)$$

dengan

Y_t : data simulasi
 T_t : komponen tren
 M_t : komponen musiman
 L_t : efek variasi kalender
 N_t : *noise*

Kajian simulasi dilakukan beracuan pada karakter data *outflow* 2003 hingga 2016 agar karakteristik data simulasi menyerupai data empiris. Kajian simulasi dilakukan melalui langkah-langkah sebagai berikut

1. Membangkitkan 2 skenario data simulasi dengan tahapan sebagai berikut:
 - a. Membangkitkan komponen tren dengan persamaan $T_t = \varphi t$ dengan ketentuan nilai koefisien tren konstan yaitu $\varphi = 0,1$
 - b. Membangkitkan komponen musiman dengan pembobotan yang berbeda-beda untuk setiap bulan dengan ketentuan sebagai berikut

$$M_t = \gamma_1 M_{1,t} + \gamma_2 M_{2,t} + \dots + \gamma_{12} M_{12,t} \quad (3.2)$$

dengan parameter musiman

$$\begin{array}{llll} \gamma_1 = 20 & \gamma_4 = 23 & \gamma_7 = 10 & \gamma_{10} = 7 \\ \gamma_2 = 23 & \gamma_5 = 20 & \gamma_8 = 7 & \gamma_{11} = 10 \\ \gamma_3 = 25 & \gamma_6 = 15 & \gamma_9 = 5 & \gamma_{12} = 15 \end{array}$$

- c. Memberikan efek kalender variasi. *Inflow* cenderung mengalami peningkatan pada bulan setelah hari raya, sedangkan *outflow* cenderung lebih tinggi pada bulan sebelum hari raya karena kebutuhan uang masyarakat pada hari raya lebih tinggi. Data simulasi dibangkitkan mengikuti pola data *outflow* untuk penentuan efek variasi kalender. Nilai variabel *dummy* untuk bulan terjadinya hari raya dan bulan sebelum hari raya.

$$L_t = 56L_{1,t-1} + 42L_{2,t-1} + 34L_{3,t-1} + 30L_{4,t-1} + 23L_{1,t} + 37L_{2,t} + 44L_{3,t} + 48L_{4,t} \quad (3.3)$$

$$L_{g,t} = \begin{cases} 1, & \text{untuk Idul Fitri pada minggu ke-}g \text{ bulan ke-}t, \\ & \text{dengan } g=1,2,3,4 \\ 0, & \text{Lainnya} \end{cases}$$

$$L_{g,t-1} = \begin{cases} 1, & \text{untuk bulan sebelum Idul Fitri (bulan ke-}t) \\ & \text{pada minggu ke-}g, \text{ dengan } g=1,2,3,4 \\ 0, & \text{Lainnya} \end{cases}$$

- d. Membangkitkan komponen *noise* yang telah *white noise* dan berdistribusi normal dengan rata-rata 0 dan varians 1.

Komponen *white noise* ditulis sebagai $a_t \sim IIDN(0,1)$.

Selanjutnya a_t digunakan untuk membentuk *noise* yang mengikuti AR(1) dan ESTAR(1) yaitu

$$N_{1,t} = 0,45N_{1,t-1} + a_t \quad (3.4)$$

$$N_{2,t} = 6,5N_{t-1} \cdot \exp(-0,25N_{t-1}^2) + a_t \quad (3.5)$$

2. Mendapatkan data simulasi melalui persamaan (3.1) dengan 2 skenario berbeda. Skenario pertama menggunakan *noise* linier sedangkan skenario ke-2 menggunakan *noise* non-linier.
3. Memodelkan data simulasi menggunakan ARIMAX
 - a. Melakukan regresi *time series* dan menyimpan residual
 - b. Menguji apakah residual telah *white noise*. Jika sudah, maka model TSR dipilih sebagai model terbaik, jika belum dilanjutkan pada Langkah 3c
 - c. Mengidentifikasi *order* ARIMAX melalui ACF dan PACF residual TSR
 - d. Melakukan estimasi parameter dan melakukan uji signifikansi parameter model ARIMAX
 - e. Melakukan cek diagnosis dan memilih model terbaik beserta *fits* dan *forecast*-nya
4. Memodelkan data simulasi menggunakan DNN
 - a. Membagi data menjadi *training* dan *testing*
 - b. Melakukan *preprocessing* dengan menstandarisasi data
 - c. Menentukan input. Input yang digunakan adalah input variabel *dummy* seperti dalam Tabel 3.4 dan input lag. Pemilihan variabel input lag dilakukan dengan 2 cara yaitu:
 - i. Menggunakan lag PACF signifikan dari data asli. Pemodelan ini selanjutnya dituliskan sebagai DNN-1.
 - ii. Menggunakan orde AR sesuai dengan model ARIMAX yang diperoleh pada Langkah 2. Jika model ARIMAX yang diperoleh pada Langkah 2 tidak memiliki orde AR ($p = 0$) maka input dipilih berdasarkan lag PACF yang signifikan dari residual TSR. Pemilihan input lag melalui cara ini sesuai dengan yang pernah dilakukan Crone (2009). Pemodelan ini selanjutnya dituliskan sebagai DNN-2.

- d. Memilih jumlah neuron optimum pada setiap *hidden layer* berdasarkan RMSEP terkecil dengan mencobakan 1 neuron hingga 10 neuron pada setiap *hidden layer*.
- e. Mendapatkan arsitektur DNN terbaik beserta *fits* dan *forecast*-nya
5. Melakukan pemodelan dengan SSA-DNN dengan tahapan sebagai berikut
 - a. Melakukan *embedding*. Pembentukan matriks lintasan dilakukan dengan *embedding* pada data deret waktu yang telah dibangkitkan menggunakan persamaan (2.31) dan (2.32)
 - b. Melakukan SVD sesuai persamaan (2.33)
 - c. Melakukan *eigentriple grouping* sesuai persamaan (2.34)
 - d. Melakukan *diagonal averaging* sesuai persamaan (2.35)
 - e. Melakukan analisis tren untuk komponen tren hasil SSA secara agregat.
 - f. Melakukan analisis regresi *dummy* untuk komponen musiman hasil SSA secara agregat.
 - g. Menjumlahkan residual tren analisis dan residual regresi *dummy* dengan komponen *noise* SSA untuk dimodelkan menggunakan DNN disertai *preprocessing*.
 - h. Menggunakan *lag* yang signifikan pada PACF dan variabel *dummy* pada Tabel 3.4 sebagai input.
 - i. Memilih jumlah neuron optimum pada setiap *hidden layer* berdasarkan RMSEP terkecil dengan mencobakan 1 neuron hingga 10 neuron pada setiap *hidden layer*.
 - j. Mendapatkan *fits* dan *forecast* SSA-DNN dengan cara menjumlahkan *fits* dan *forecast* dari *trend analysis*, regresi *dummy* dan DNN.
6. Membandingkan ARIMAX, DNN, SSA-DNN pada setiap skenario data simulasi dan menarik kesimpulan.

3.2 Kajian Terapan

Data yang digunakan dalam kajian terapan penelitian ini adalah data sekunder yang diperoleh dari Bank Indonesia. Data yang digunakan terdiri dari data *inflow* dan *outflow* uang kartal secara nasional untuk setiap pecahan mata uang, mulai dari pecahan terbesar Rp.100.000,- sampai pecahan terkecil Rp.1000,- rupiah. Periode data yang digunakan adalah mulai dari Januari 2003 hingga Desember 2016. Data penelitian ini kemudian dibagi menjadi data *training* mulai Januari 2003 hingga Desember 2015, serta data *testing* mulai Januari 2016 hingga Desember 2016. Variabel yang digunakan ditunjukkan dalam Tabel 3.1.

Tabel 3.1. Variabel Penelitian

Data	Variabel	Keterangan
<i>Outflow</i>	$Y_{1,t}$	pecahan Rp100.000,- pada bulan ke- t
	$Y_{2,t}$	pecahan Rp50.000,- pada bulan ke- t
	$Y_{3,t}$	pecahan Rp20.000,- pada bulan ke- t
	$Y_{4,t}$	pecahan Rp10.000,- pada bulan ke- t
	$Y_{5,t}$	pecahan Rp5.000,- pada bulan ke- t
	$Y_{6,t}$	pecahan Rp2.000,- pada bulan ke- t
	$Y_{7,t}$	pecahan Rp1.000,- pada bulan ke- t
<i>Inflow</i>	$Y_{8,t}$	pecahan Rp100.000,- pada bulan ke- t
	$Y_{9,t}$	pecahan Rp50.000,- pada bulan ke- t
	$Y_{10,t}$	pecahan Rp20.000,- pada bulan ke- t
	$Y_{11,t}$	pecahan Rp10.000,- pada bulan ke- t
	$Y_{12,t}$	pecahan Rp5.000,- pada bulan ke- t
	$Y_{13,t}$	pecahan Rp2.000,- pada bulan ke- t
	$Y_{14,t}$	pecahan Rp1.000,- pada bulan ke- t

sedangkan data yang digunakan dalam kajian terapan penelitian ini meliputi variabel yang ada dalam Tabel 3.2 dengan struktur yang digambarkan dalam Tabel 3.3. Menurut Suhartono dkk (2017) Dummy variasi kalender yang digunakan untuk *inflow* adalah $L_{g,t}$ dan $L_{g,t+1}$ sedangkan untuk *outflow* adalah $L_{g,t-1}$ dan $L_{g,t}$. Hal ini juga didukung hasil eksplorasi data dan penelitian sebelumnya tentang *inflow* dan *outflow*. Penjelasan notasi variabel *dummy* ditampilkan dalam Tabel 3.4.

Tabel 3.3. Struktur Data

Data	Tahun	Bulan	Outflow	Inflow
			$Y_{h,t} (h=1,2,...,7)$	$Y_{h,t} (h=8,9,...,14)$
Training	2003	1	$Y_{h,1}$	$Y_{h,1}$
	2003	2	$Y_{h,2}$	$Y_{h,2}$
	\vdots	\vdots	\vdots	\vdots
	2003	12	$Y_{h,12}$	$Y_{h,12}$
	2004	1	$Y_{h,13}$	$Y_{h,13}$
	2004	2	$Y_{h,14}$	$Y_{h,14}$
	\vdots	\vdots	\vdots	\vdots
	2015	11	$Y_{h,155}$	$Y_{h,155}$
Testing	2015	12	$Y_{h,156}$	$Y_{h,156}$
	2016	1	$Y_{h,157}$	$Y_{h,157}$
	2016	2	$Y_{h,158}$	$Y_{h,158}$
	\vdots	\vdots	\vdots	\vdots
	2016	12	$Y_{h,168}$	$Y_{h,168}$

Variabel *dummy* yang digunakan dalam penelitian ini ditunjukkan dalam Tabel 3.4.

Tabel 3.4. Variabel *Dummy*

Variabel <i>dummy</i>	Keterangan
Tren	t , dengan $t=1,2,...,n$
Musiman	$M_{j,t} = \begin{cases} 1, & \text{untuk bulan ke-}j. j=1, 2, ..., 12 \\ 0, & \text{lainnya} \end{cases}$ j bersesuaian dengan bulan kalender Masehi
Variasi Kalender	$L_{g,t} = \begin{cases} 1 & \text{untuk Idul Fitri minggu ke-}g \text{ bulan ke-}t, \\ & \text{dengan } g=1,2,3,4 \\ 0 & \text{Lainnya} \end{cases}$
	$L_{g,t-1} = \begin{cases} 1 & \text{untuk bulan sebelum bulan Idul Fitri} \\ & \text{pada minggu ke-}g, \text{ dengan } g=1,2,3,4 \\ 0 & \text{Lainnya} \end{cases}$
	$L_{g,t+1} = \begin{cases} 1 & \text{untuk bulan sesudah bulan Idul Fitri} \\ & \text{pada minggu ke-}g, \text{ dengan } g=1,2,3,4 \\ 0 & \text{lainnya} \end{cases}$

Data *time series* di bidang ekonomi sering disusun dalam periode bulanan sehingga sering memiliki pola musiman. Permintaan uang yang mempengaruhi *outflow* dan *inflow* di Indonesia banyak dipengaruhi oleh hari raya keagamaan seperti Idul Fitri (Ahmad, Setiawan, Suhartono, & Masun, 2015). Sedangkan dalam penelitian ini berfokus pada efek variasi kalender yang terjadi karena Hari Raya Idul Fitri dengan rincian variabel pada Tabel 3.4.

Berikut adalah langkah analisis yang dilakukan untuk mencapai tujuan yang diharapkan dalam penelitian ini sesuai permasalahan yang telah dirumuskan pada data empiris.

1. Melakukan analisis statistika deskriptif pada setiap pecahan untuk mengetahui gambaran karakteristik data *inflow* dan *outflow*, dengan langkah sebagai berikut:
 - a. Melakukan perhitungan statistika deskriptif untuk data *inflow* dan *outflow*, meliputi *mean*, median, standar deviasi, *skewness*, dan *kurtosis*.
 - b. Melakukan identifikasi pola data *inflow* dan *outflow* melalui plot *time series*, *boxplot* atau visualisasi lain yang dapat menunjukkan indikasi tren, musiman, variasi kalender, dan *outlier*. Berdasarkan plot *time series*, diasumsikan bahwa data memiliki pola tren linier, musiman, variasi kalender, dan residual yang belum acak.
2. Memodelkan data *inflow outflow* menggunakan ARIMAX
 - a. Melakukan regresi *time series* dan menyimpan residual dengan variabel prediktor dalam Tabel 3.4 dan ditambahkan dengan 4 variabel *dummy* yaitu 2 variabel *dummy* untuk membedakan data sebelum tahun 2007 ($d_1=1$), 2007-2010 ($d_2=1$) dan setelah 2011 ($d_1= d_2=0$) serta 2 variabel *dummy* interaksi d dengan t . Penggunaan *dummy* tambahan didasarkan pada hasil eksplorasi data dan bertujuan agar model mampu menangkap perubahan tren yang terjadi. Selain itu ARIMAX dilakukan tanpa intersep agar lebih mudah diinterpretasikan.

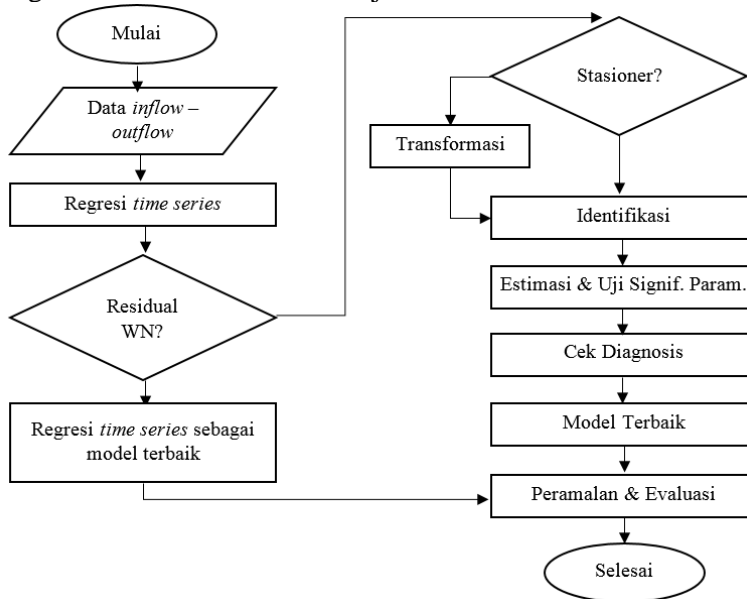
- b. Menguji apakah residual telah *white noise*. Jika sudah, maka model TSR dipilih sebagai model terbaik, jika belum dilanjutkan pada Langkah 2c
 - c. Mengidentifikasi *order* ARIMAX melalui ACF dan PACF residual TSR
 - d. Melakukan estimasi parameter dan melakukan uji signifikansi parameter model ARIMAX
 - e. Melakukan cek diagnosis dan memilih model terbaik beserta *fits* dan *forecast*-nya
3. Memodelkan data *inflow* dan *outflow* menggunakan DNN
 - a. Membagi data menjadi *training* dan *testing*
 - b. Melakukan *preprocessing* dengan menstandarisasi data
 - c. Menentukan input. Input yang digunakan adalah input variabel *dummy* seperti dalam Tabel 3.4 dan input lag. Pemilihan variabel input lag dilakukan dengan 2 cara yaitu:
 - i. Menggunakan lag PACF signifikan dari data asli. Pemodelan ini selanjutnya dituliskan sebagai DNN-1.
 - ii. Menggunakan orde AR sesuai dengan model ARIMAX yang diperoleh pada Langkah 2. Jika model ARIMAX yang diperoleh pada Langkah 2 tidak memiliki orde AR ($p = 0$) maka input dipilih berdasarkan lag PACF yang signifikan dari residual TSR. Pemilihan input lag melalui cara ini sesuai dengan yang pernah dilakukan Crone (2009). Pemodelan ini selanjutnya dituliskan sebagai DNN-2.
 - f. Memilih jumlah neuron optimum pada setiap *hidden layer* berdasarkan RMSEP terkecil dengan mencoba 1 neuron hingga 10 neuron pada setiap *hidden layer*.
 - g. Mendapatkan arsitektur DNN terbaik beserta *fits* dan *forecast*-nya dengan melakukan *post-processing* pada hasil DNN.
3. Membentuk model *hybrid* SSA-DNN dengan tahapan sebagai berikut

- a. Melakukan *embedding*. Pembentukan matriks lintasan dilakukan dengan *embedding* pada data deret waktu yang telah dibangkitkan menggunakan persamaan (2.31) dan (2.32)
 - b. Melakukan SVD sesuai persamaan (2.33)
 - c. Melakukan *eigentriple grouping* sesuai persamaan (2.34)
 - d. Melakukan *diagonal averaging* sesuai persamaan (2.35)
 - e. Melakukan analisis tren untuk komponen tren hasil SSA secara agregat menggunakan regresi polinomial hingga orde 3.
 - f. Melakukan analisis regresi *dummy* untuk komponen musiman hasil rekonstruksi SSA secara agregat sesuai variabel pada Tabel 3.4 dengan ditambahkan 11 variabel interaksi antara *dummy* musiman dengan t . Hal ini dilakukan karena adanya pola musiman multiplikatif (melebar) sepanjang t pada data komponen musiman hasil rekonstruksi SSA
 - g. Menjumlahkan residual analisis tren dan residual regresi *dummy* dengan komponen noise SSA untuk dimodelkan menggunakan DNN disertai *preprocessing*.
 - h. Menentukan variabel yang akan digunakan dalam DNN berdasarkan lag yang signifikan pada PACF dan variabel *dummy* pada Tabel 3.4.
 - i. Memilih jumlah neuron optimum pada setiap hidden layer berdasarkan RMSEP terkecil dengan mencoba 1 neuron hingga 10 neuron pada setiap hidden layer.
 - j. Mendapatkan arsitektur DNN terbaik sehingga didapatkan *fits* dan *forecast* DNN.
 - k. Mendapatkan *fits* dan *forecast* SSA-DNN dengan cara menjumlahkan *fits* dan *forecast* dari *trend analysis*, regresi *dummy* dan DNN.
4. Mengevaluasi dan membandingkan performa model ARIMAX, DNN dan *hybrid* SSA-DNN dalam memodelkan *inflow* dan *outflow* uang kartal di Indonesia menggunakan

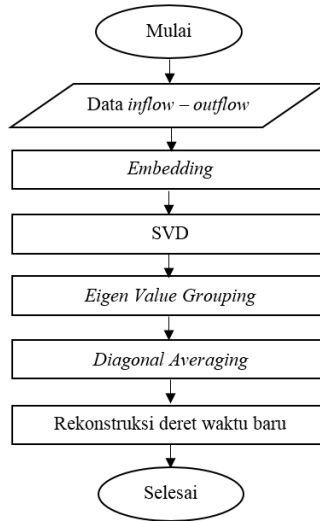
kriteria RMSEP untuk 3 periode, 6 periode, 9 periode dan 12 periode.

5. Melakukan peramalan *inflow* dan *outflow* uang kartal di Indonesia periode tahun 2018 untuk masing-masing pecahan mata uang berdasarkan model terbaik yang terbentuk.
6. Menarik kesimpulan berdasarkan hasil analisis dan pembahasan serta memberikan saran yang bersesuaian. Selain itu juga menambahkan hasil penelitian sebelumnya yang bersesuaian dengan hasil analisis untuk memperkuat kesimpulan yang diambil.

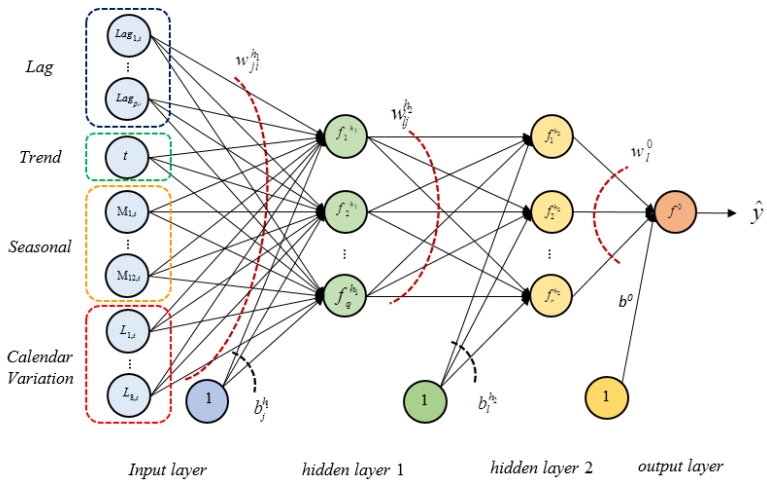
Langkah analisis tersebut digambarkan dalam diagram alir Gambar 3.1 hingga Gambar 3.4. Tahapan ARIMAX diilustrasikan dalam Gambar 3.1, tahapan SSA ditunjukkan pada Gambar 3.2, skema model DNN diilustrasikan dalam Gambar 3.3, sedangkan pembentukan model SSA-DNN telah dijelaskan pada Gambar 2.3. Diagram Alir secara umum ditunjukkan dalam Gambar 3.4.



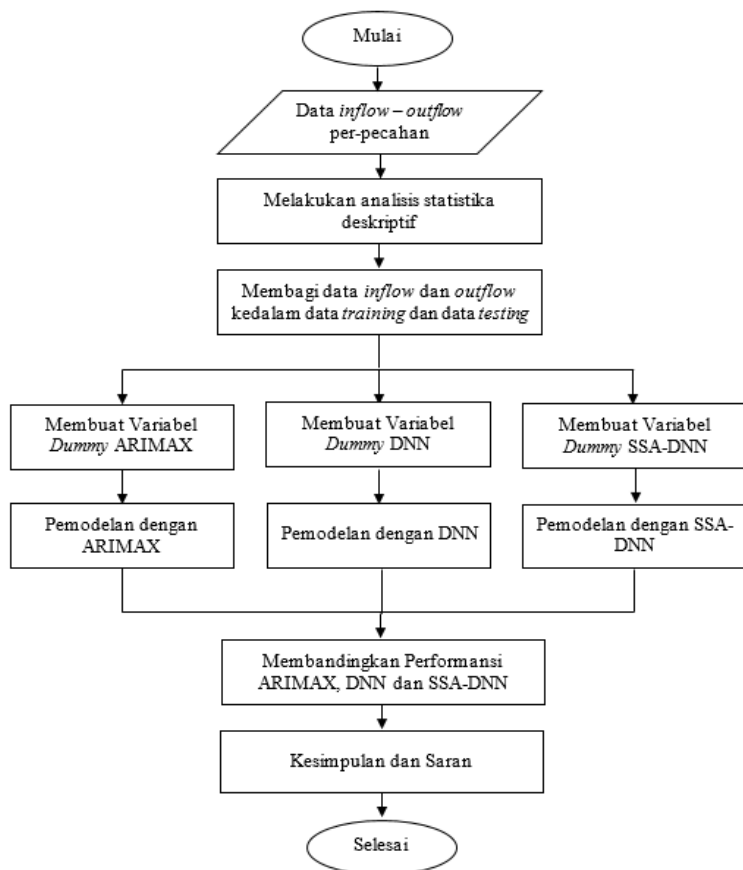
Gambar 3.1. Tahapan ARIMAX



Gambar 3.2. Tahapan SSA



Gambar 3.3. Skema DNN

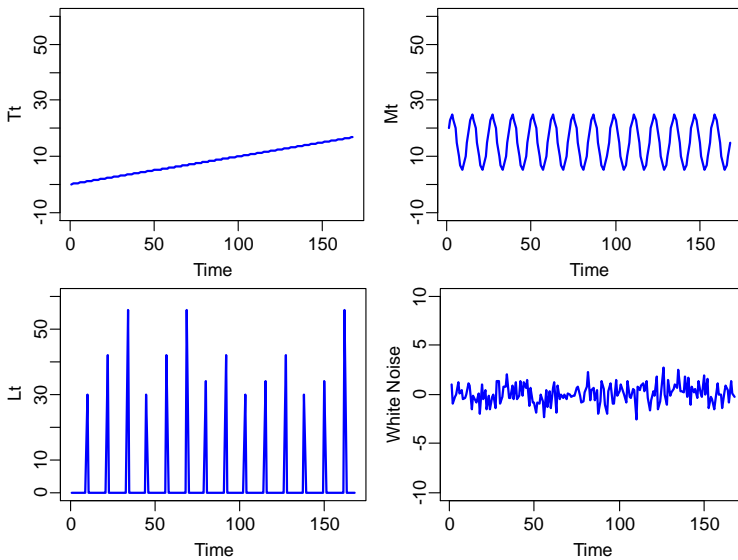


Gambar 3.4. Diagram Alir

BAB IV ANALISIS DAN PEMBAHASAN

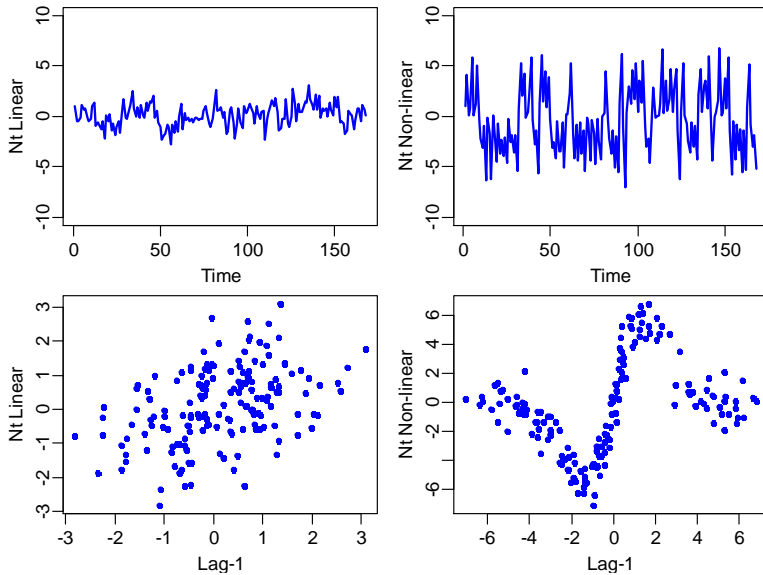
4.1 Kajian Simulasi

Kajian simulasi bertujuan untuk mengetahui performa metode ARIMAX, DNN dan SSA-DNN dalam melakukan peramalan. Kajian dilakukan dengan membangkitkan data yang mengandung *trend*, musiman, variasi kalender dan *noise*. Skenario pertama menggunakan *noise* yang berpola linier yaitu AR(1) sedangkan skenario ke-2 menggunakan *noise* yang mengikuti model ESTAR(1). Gambar 4.1 menunjukan pola dari *trend*, musiman, variasi kalender dan *noise* yang dibangkitkan.



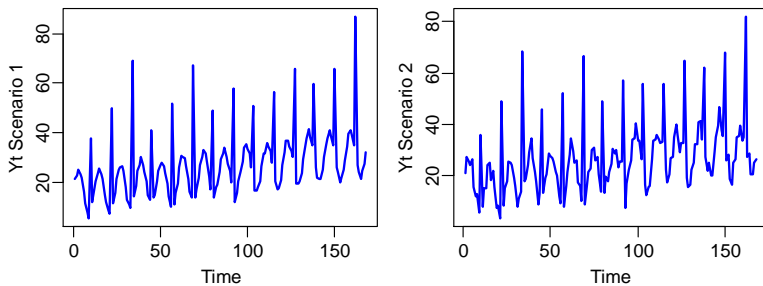
Gambar 4.1. Pola Komponen Data Simulasi

Gambar 4.2 (a) menunjukkan pola *noise* linier yang mengikuti AR(1). *Noise* ini memiliki korelasi linier dengan *lag*-1 sehingga apabila diplotkan dengan *lag*-1 akan tampak seperti Gambar 4.2 (c). Sedangkan Gambar 4.2 (b) menunjukkan *noise* yang mengikuti ESTAR(1) dengan pola seperti Gambar 4.2 (d)



Gambar 4.2. Pola Komponen Noise Data Simulasi

Komponen *trend*, musiman, variasi kalender dan *noise* digunakan untuk menyusun data simulasi skenario 1 dan skenario 2 mengikuti persamaan (3.1) hingga (3.5) sehingga menghasilkan data seperti pada Gambar 4.3. Setiap skenario direplikasi sebanyak 5 kali untuk dicobakan pada setiap metode. Antar replikasi dibedakan oleh komponen random a_t yang dibangkitkan secara terpisah untuk masing-masing replikasi.



Gambar 4.3. Pola Data Simulasi Skenario 1 dan Skenario 2

4.1.1 Skenario 1 dengan *Noise* mengikuti AR(1)

Skenario 1 akan dimodelkan menggunakan metode ARIMAX, DNN-1, DNN-2, dan *hybrid* SSA-DNN dan dibandingkan berdasarkan RMSE dan sMAPE. Perbedaan antara DNN-1 dan DNN-2 adalah pada pemilihan variabel lag sebagai input. Pada DNN-1 lag yang digunakan adalah lag PACF signifikan dari data asli. Sedangkan pada DNN-2 lag yang digunakan mengikuti orde AR model ARIMAX.

a. Peramalan Skenario 1 dengan ARIMAX

Time series regression dilakukan pada data simulasi skenario 1 sehingga menghasilkan residual yang akan dimodelkan menggunakan ARIMA. Pada skenario 1 replikasi 1, hasil estimasi regresi *time series* ditunjukkan dalam Table 4.1

Tabel 4.1. TSR $Y_{t-1,1}$

Prediktor	Estimasi	SE	T	P
t	0,11	0,00	53,48	<0,001
$M_{1,t}$	19,68	0,33	58,88	<0,002
$M_{2,t}$	22,54	0,34	67,27	<0,003
$M_{3,t}$	24,50	0,34	72,93	<0,004
$M_{4,t}$	22,49	0,34	66,77	<0,005
$M_{5,t}$	19,74	0,34	58,45	<0,006
$M_{6,t}$	14,97	0,34	43,60	<0,007
$M_{7,t}$	9,76	0,36	27,43	<0,008
$M_{8,t}$	6,57	0,36	18,43	<0,009
$M_{9,t}$	4,77	0,36	13,08	<0,010
$M_{10,t}$	6,58	0,38	17,41	<0,011
$M_{11,t}$	9,56	0,36	26,59	<0,012
$M_{12,t}$	14,36	0,34	41,70	<0,013
$L_{1,t-1}$	56,97	0,81	70,32	<0,014
$L_{2,t-1}$	41,50	0,57	72,22	<0,015
$L_{3,t-1}$	34,48	0,66	52,07	<0,016
$L_{4,t-1}$	30,33	0,57	53,02	<0,017
$L_{1,t}$	0,42	0,81	0,52	0,602
$L_{2,t}$	-1,34	0,57	-2,35	0,020
$L_{3,t}$	0,77	0,66	1,16	0,247
$L_{4,t}$	1,09	0,57	1,90	0,059

Setelah regresi *time series* dilakukan, residual yang dihasilkan kemudian dimodelkan menggunakan ARIMA. Tahapan identifikasi dilakukan dengan bantuan ACF dan PACF dalam Lampiran yang menunjukkan ACF dengan pola *dies down* dan PACF dengan pola *cuts off* setelah lag-1 dan musiman lag-12. Sehingga kemungkinan model ARIMA yang dapat dibentuk adalah ARIMA(1,0,0) atau ARIMA (1,0,0) (1,0,0)¹². Tabel 4.2 menunjukkan perbandingan antara kedua kemungkinan model.

Tabel 4.2. Kemungkinan Model ARIMA Residual TSR Yt_1_1

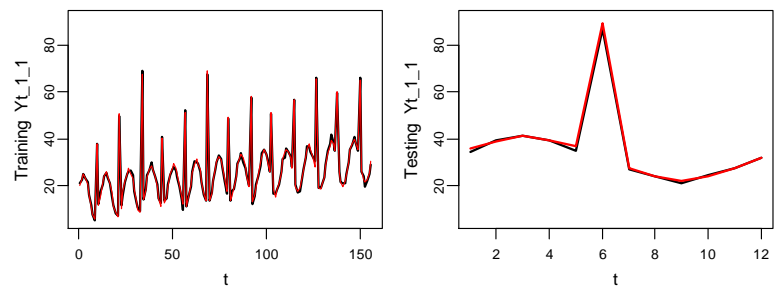
Model	AIC
ARIMA(1,0,0)	434,52
ARIMA(1,0,0)(1,0,0)12	437,45

Tabel 4.2 menunjukkan bahwa model tanpa parameter musiman memiliki AIC lebih kecil sehingga model ARIMA(1,0,0) dipilih sebagai model terbaik. Estimasi parameter model ARIMA (1,0,0) ditunjukkan pada Tabel 4.3

Tabel 4.3. Estimasi Model ARIMA Residual TSR Yt_1_1

Parameter	Estimasi	SE	T	P
Intersep	-0,077	0,114	-0,674	0,251
AR1	0,332	0,077	4,316	<0,001

Berdasarkan model tersebut, diperoleh nilai *fits* dan *forecast* untuk 12 periode yang ditampilkan dalam Gambar 4.5



Gambar 4.4. Fits dan Forecast ARIMAX Yt_1_1

Gambar 4.4 menunjukkan bahwa pola *trend*, musiman dan variasi kalender pada data simulasi skenario 1 replikasi 1 dapat ditangkap dengan baik oleh model ARIMAX. Hal tersebut telah sesuai karena data simulasi dibangkitkan menggunakan model

ARIMAX. Pemodelan ARIMAX dilakukan untuk replikasi ke-2 sampai ke-5 pada skenario 1 sehingga diperoleh model terbaik ARIMAX untuk setiap replikasi dalam persamaan

a. Persamaan ARIMAX skenario 1 replikasi 1

$$Y_{1,1,t} = (1 - 0,332Y_{t-1})^{-1}a_t + 0,106t + 19,682M_{1,t} + \dots + 1,089L_{4,t}$$

b. Persamaan ARIMAX skenario 1 replikasi 2

$$Y_{1,2,t} = (1 - 0,492Y_{t-1})^{-1}a_t + 0,103t + 20,184M_{1,t} + \dots - 0,996L_{4,t}$$

c. Persamaan ARIMAX skenario 1 replikasi 3

$$Y_{1,3,t} = (1 - 0,439Y_{t-1})^{-1}a_t + 0,101t + 19,583M_{1,t} + \dots - 0,317L_{4,t}$$

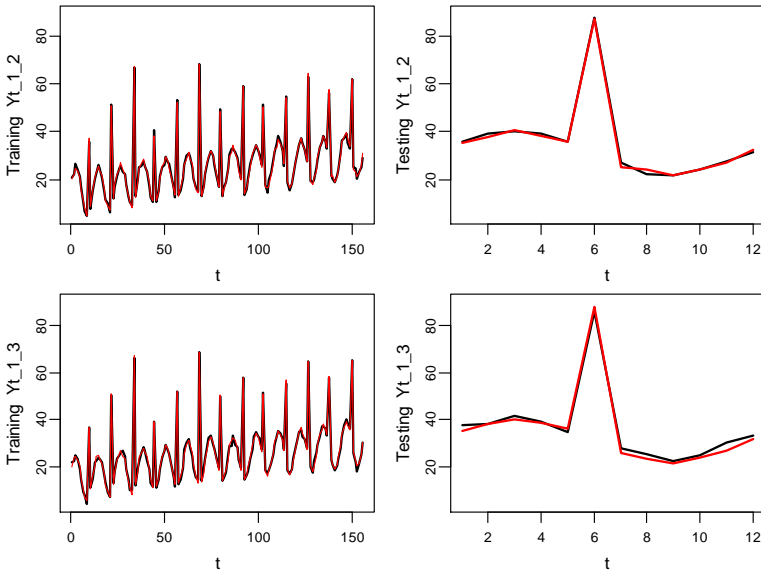
d. Persamaan ARIMAX skenario 1 replikasi 4

$$Y_{1,4,t} = (1 - 0,473Y_{t-1})^{-1}a_t + 0,099t + 19,748M_{1,t} + \dots + 0,459L_{4,t}$$

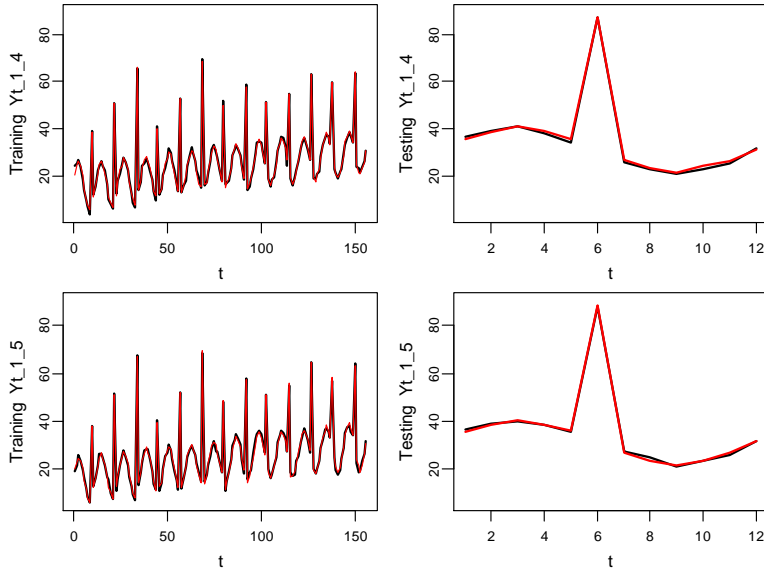
e. Persamaan ARIMAX skenario 1 replikasi 5

$$Y_{1,5,t} = (1 - 0,422Y_{t-1})^{-1}a_t + 0,099t + 22,886M_{1,t} + \dots + 0,994L_{4,t}$$

Berdasarkan model terbaik setiap replikasi, diperoleh *fits* dan *forecast* ARIMAX yang ditunjukkan dalam Gambar 4.5.



Gambar 4.5 Fits dan Forecast ARIMAX Skenario 1

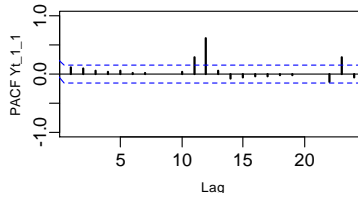


Gambar 4.5. (Lanjutan) Fits dan Forecast ARIMAX Skenario 1

Gambar 4.5 menunjukkan bahwa performa ARIMAX pada kelima replikasi terlihat baik dan konsisten mampu mengikuti pola data. Residual yang dihasilkan model ARIMAX juga telah bersifat *white noise*. Namun perhitungan akurasi perlu dilakukan untuk membandingkan performa ARIMAX dengan DNN dan SSA-DNN.

b. Peramalan Skenario 1 dengan DNN-1

Preprocessing DNN-1 dilakukan dengan menstandarisasi data. Tahap pertama dalam peramalan menggunakan DNN-1 adalah memilih variabel input. Pemilihan input dilakukan berdasarkan lag-PACF yang signifikan. Selain itu variabel *dummy* musiman dan variasi kalender juga ditambahkan sebagai input untuk memperkuat DNN dalam menangkap variasi kalender.



Gambar 4.6. PACF Y_{t-1}

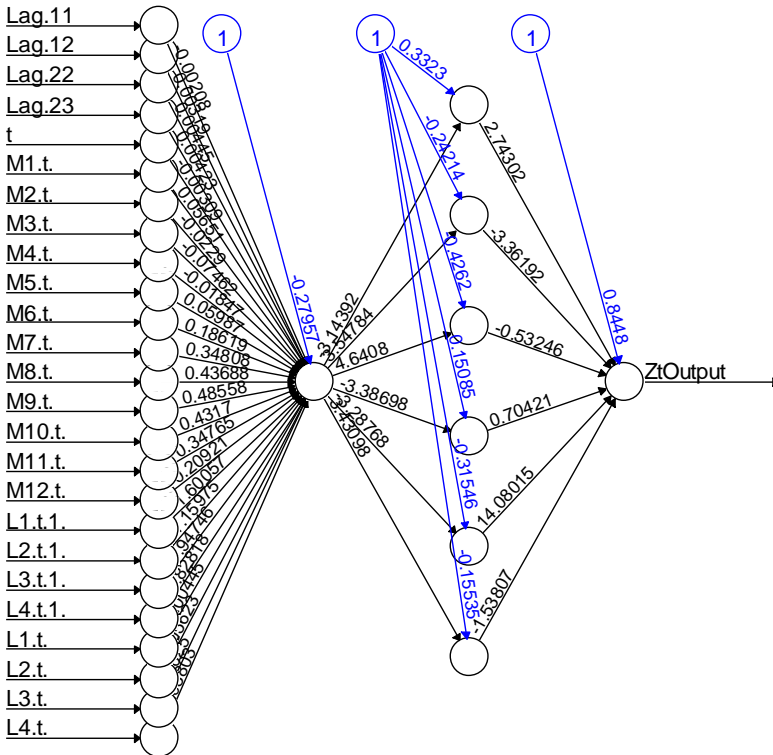
Gambar 4.6 menunjukkan PACF dari data skenario 1 replikasi 1 pada lag-11, 12, 22 dan lag-23 signifikan sehingga keempat lag-tersebut dijadikan input DNN-1 bersama variabel *dummy*. Pemilihan jumlah *neuron* pada setiap *hidden layer* dilakukan dengan *trial and error* yaitu mencoba sebanyak 1 sampai 10 *neuron* pada setiap *hidden layer* sehingga terdapat 100 kombinasi. Setiap kombinasi dilakukan 10 kali percobaan (ulangan dengan *set seed*) berbeda untuk mengantisipasi kemungkinan estimasi tidak konvergen pada *set seed* tertentu. Diantara 10 percobaan kemudian dipilih 1 model dengan RMSE *testing* terkecil untuk mewakili kombinasi *neuron* yang bersesuaian. Tabel 4.4 menunjukkan ringkasan dari 100 model DNN-1 untuk skenario 1 replikasi 1.

Tabel 4.4. Pemilihan Jumlah Neuron DNN Y_{t-1}

i	Neuron H-1	Neuron H-2	k	RMSE	
				Training	Testing
1	1	1	1	1,016	1,541
2	2	1	3	0,841	4,352
3	3	1	3	1,212	5,065
...
51	1	6	1	1,014	1,431
...
100	10	10	1	0,435	6,593

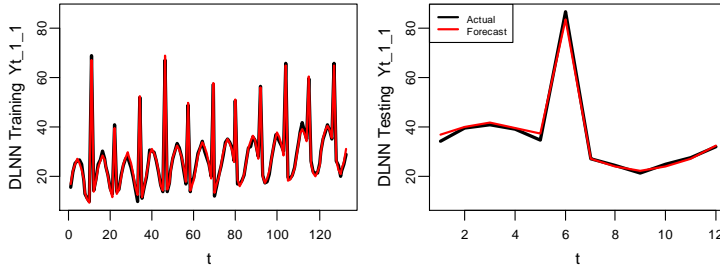
Tabel 4.4 menunjukkan bahwa model ke-51 yang menggunakan 1 dan 6 *neuron* pada *hidden layer* pertama dan ke-2 pada ulangan percobaan ke-1 (k menunjukkan ulangan percobaan ke- k) memiliki RMSE *testing* yang paling kecil, yaitu 1.431 sehingga model tersebut dipilih sebagai model terbaik dengan *neuron* optimal.

Arsitektur model DNN-1 yang terbentuk ditampilkan dalam Gambar 4.7. Angka yang terbentuk pada garis yang menghubungkan antar *neuron* menunjukkan hasil estimasi bobot antar *neuron* yang dihubungkan oleh garis penghubung. Angka dicetak biru pada garis biru menunjukan bobot dari intersep atau lebih dikenal sebagai *bias* yang.



Gambar 4.7. Arsitektur DNN-1 Y_t-1_1

Model terbaik menghasilkan nilai *fits* dan *forecast* seperti pada Gambar 4.8 yang menunjukkan bahwa metode DNN-1 dapat menangkap pola tren, musiman dan variasi kalender dengan sangat baik baik pada data *training* maupun data *testing*.



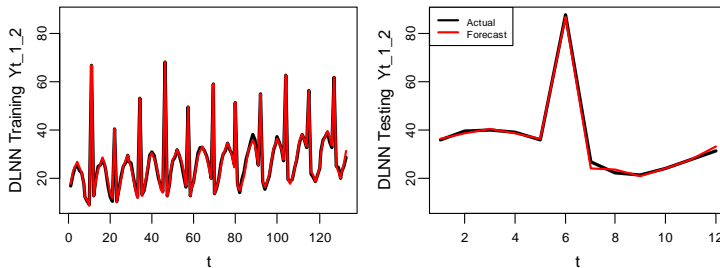
Gambar 4.8. Fits dan Forecast DNN $Y_{t_1_1}$

Pemodelan DNN-1 dilanjutkan untuk replikasi 2 hingga replikasi 5 pada skenario 2 sehingga diperoleh model terbaik pada setiap replikasi seperti pada Tabel 4.5.

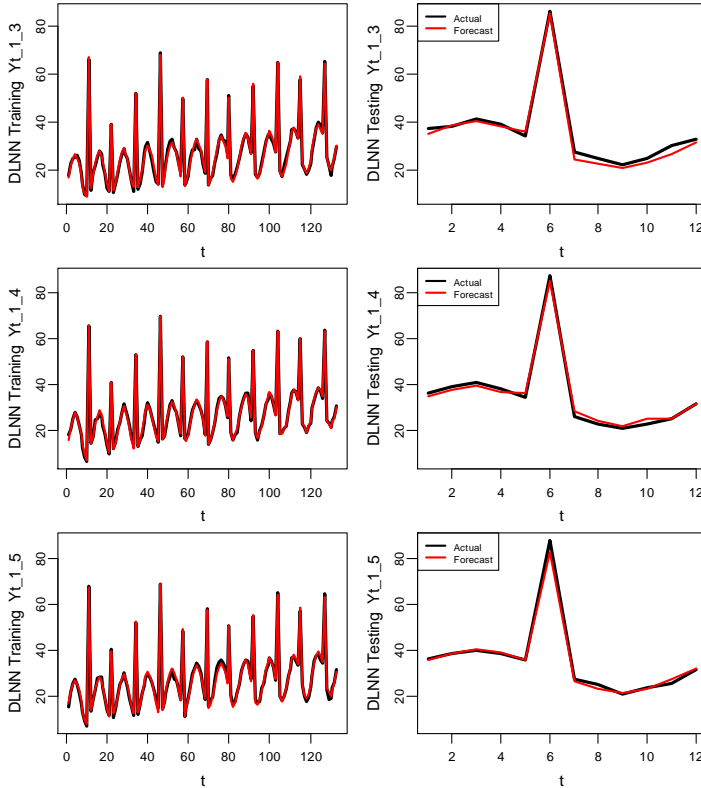
Tabel 4.5. Model DNN-1 Terbaik Skenario 1

i	<i>Neuron</i>		k	RMSE Testing
	<i>Hidden-1</i>	<i>Hidden-2</i>		
51	1	6	1	1,43
83	3	9	5	1,18
62	2	7	3	1,90
34	4	4	4	1,65
62	2	7	5	1,64

Tabel 4.5 menunjukkan model DNN-1 terbaik untuk skenario 2 replikasi ke-2 hingga ke-5 berturut-turut adalah model DNN-1 nomor 83, 62, 34 dan 62 pada percobaan *set seed* ke-5, 3, 4, dan 5. Berdasarkan model terbaik yang diperoleh, didapatkan *fits* dan *forecast* untuk replikasi 2 hingga 5 pada Gambar 4.9 yang menunjukkan bahwa DNN-1 pada kelima replikasi sangat baik dan konsisten mampu mengikuti pola data.



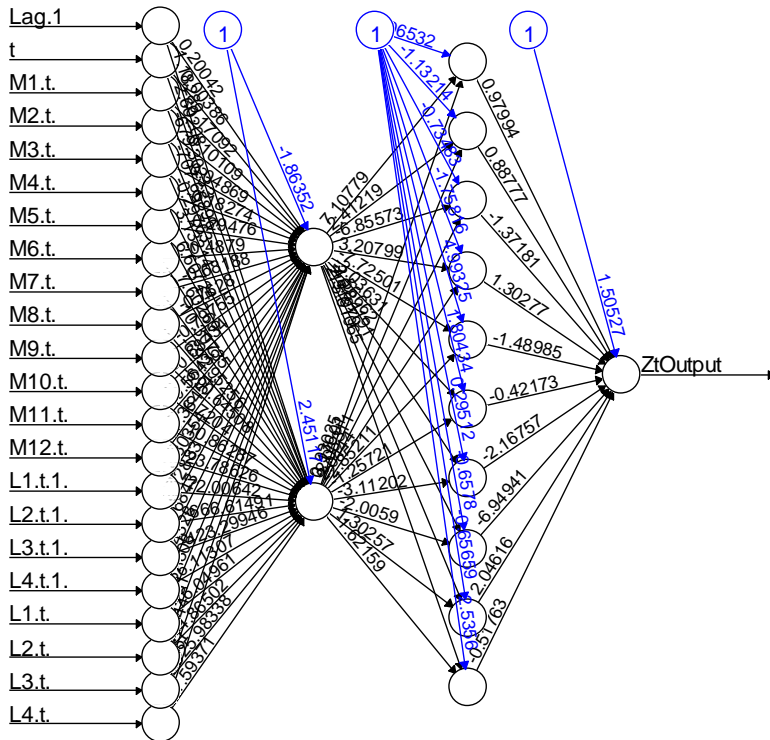
Gambar 4.9. Fits dan Forecast DNN Skenario 1



Gambar 4.9. (Lanjutan) Fits dan Forecast DNN Skenario 1

c. Peramalan Skenario 1 dengan DNN-2

Pada peramalan dengan DNN-2 *preprocessing* dilakukan dengan cara yang sama seperti DNN-1. Input yang digunakan adalah lag-1 karena semua model ARIMAX pada kelima replikasi skenario 1 memiliki orde $p = 1$. Proses pemilihan jumlah neuron optimum juga sama dengan DNN-1. Sehingga diperoleh model terbaik untuk replikasi 1 yaitu dengan 2 neuron pada *hidden layer* pertama dan 10 neuron pada *hidden layer* ke-2. Arsitektur model DNN-2 terbaik yang dihasilkan adalah sebagai berikut



Gambar 4.10. Arsitektur DNN-2 Yt_1_1

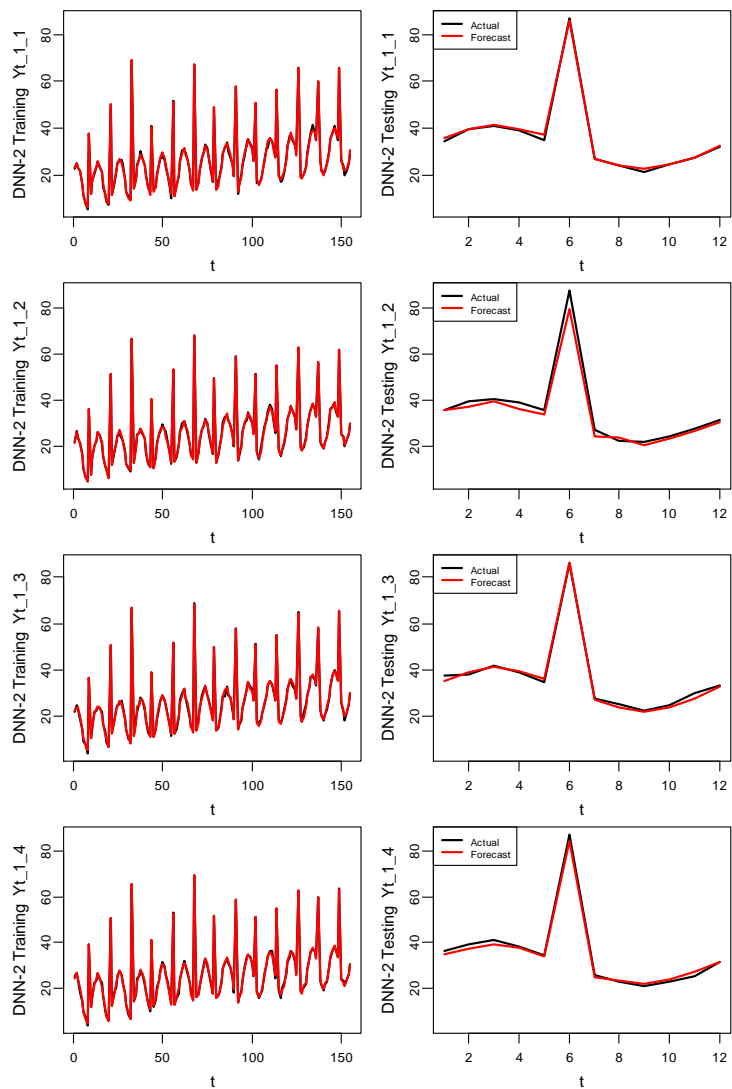
Langkah yang sama dilakukan terhadap replikasi 2 hingga 5 sehingga dihasilkan daftar model terbaik seperti dalam Tabel 4.6

Tabel 4.6. Model DNN-2 Terbaik Skenario 1

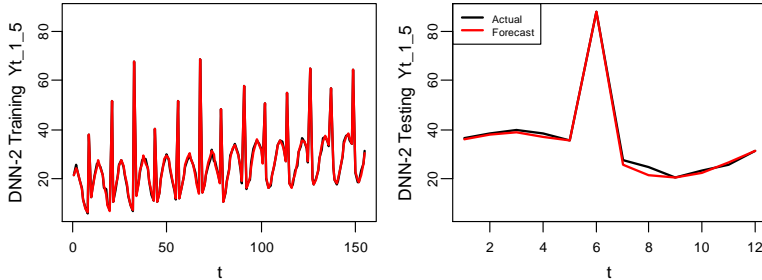
i	Neuron		k	RMSE Testing
	Hidden-1	Hidden-2		
92	2	10	2	0,96
34	4	4	4	2,89
72	2	8	2	1,24
92	2	10	1	1,36
33	3	4	3	1,28

Tabel 6 menunjukkan bahwa RMSE dari model DNN-2 secara umum telah berada di sekitar nilai 1, sesuai dengan *noise* yang

dibangkitkan.. Berdasarkan model terbaik yang diperoleh dari setiap replikasi data, hasil pemodelan ditampilkan sebagai berikut.



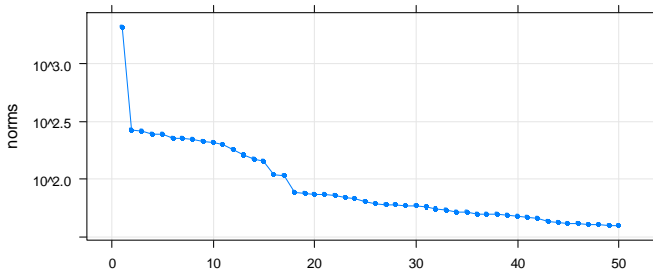
Gambar 4.11. Fits dan Forecast DNN-2 Skenario 1



Gambar 4.11. (Lanjutan) Fits dan Forecast DNN-2 Skenario 1

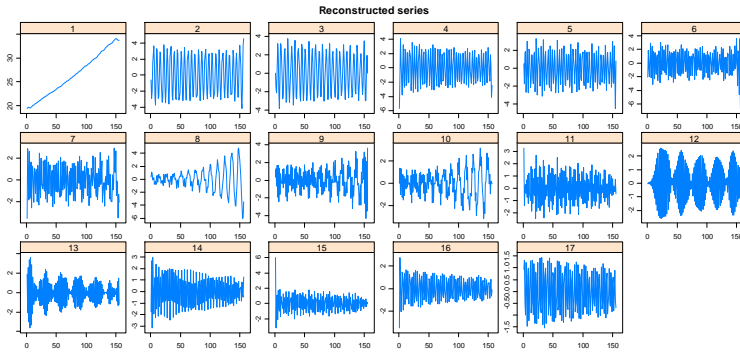
d. Peramalan Skenario 1 dengan SSA-DNN

Pemodelan menggunakan SSA-DNN dilakukan secara agregat pada setiap komponen yang terekonstruksi. Rekonstruksi dilakukan dengan mengelompokkan *eigen triples* yang berpola *trend* dan musiman menjadi 2 komponen. Selanjutnya komponen *trend* dimodelkan menggunakan regresi polinomial hingga orde 3 dengan variabel prediktor t . Sedangkan komponen musiman dimodelkan menggunakan regresi *dummy*. Pada skenario 1 replikasi 1, pengelompokan *eigen triples* dilakukan berdasarkan plot *eigenvalues* pada Gambar 4.12



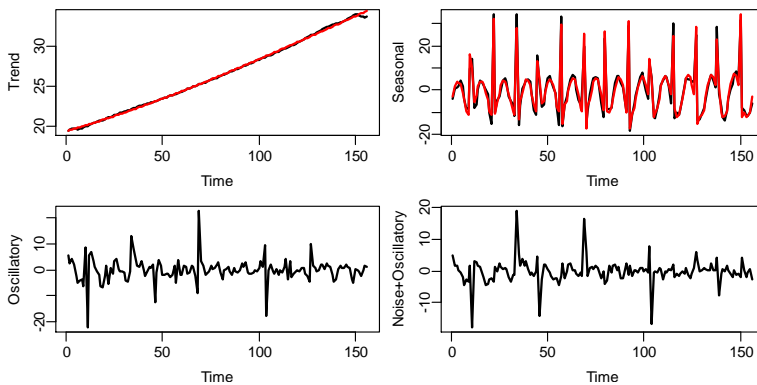
Gambar 4.12. Plot Eigenvalues Yt_1_1

Komponen *noise* diidentifikasi melalui perubahan nilai *eigenvalue* yang turun lambat. Gambar 4.12 menunjukkan penurunan lambat terjadi mulai *eigenvalue* ke-18 dan seterusnya. Sehingga *eigen triples* ke-1 hingga ke-17 dikelompokkan kedalam *trend* dan musiman. Identifikasi dilakukan dengan rekonstruksi *eigen triples* secara individu seperti pada Gambar 4.13



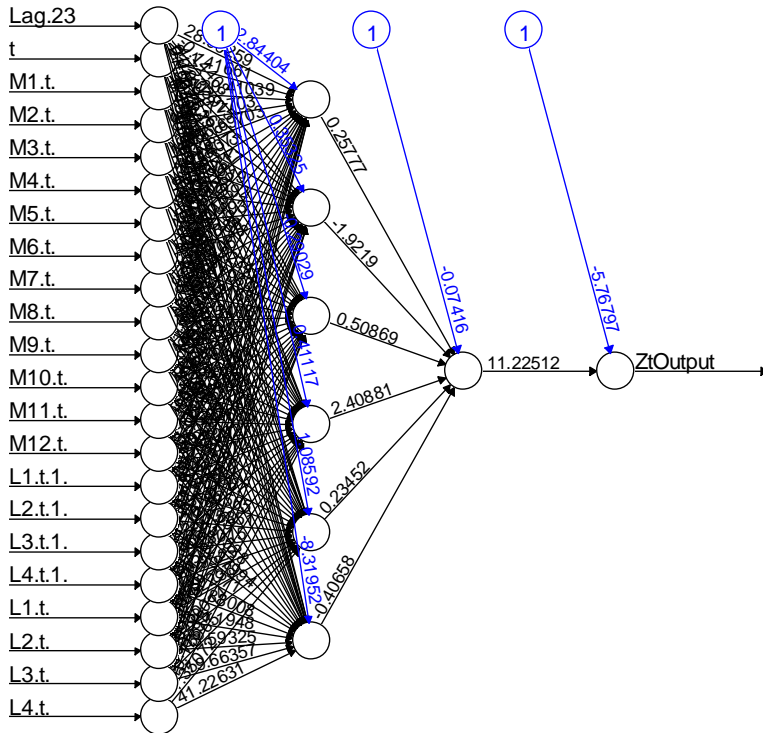
Gambar 4.13. Rekonstruksi Individu Y_{t_1}

Gambar 4.13 menunjukkan bahwa komponen pertama memiliki pola *trend* sehingga akan dimodelkan dengan regresi polinomial, sedangkan komponen ke-2 hingga ke-17 memiliki pola musiman sehingga akan direkonstruksi ulang menjadi 1 komponen dan dimodelkan menggunakan regresi *dummy*. Hasil rekonstruksi ulang dan pemodelan ditampilkan dalam Gambar 4.14



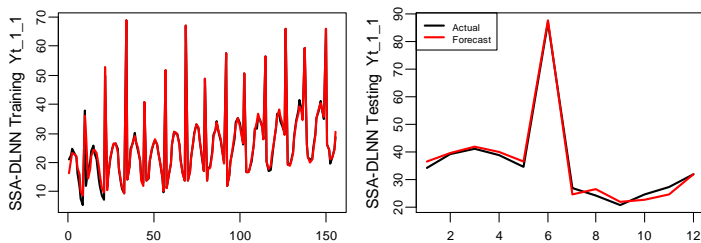
Gambar 4.14. Rekonstruksi & Pemodelan Agregat Y_{t_1}

Gambar 4.14 menunjukkan bahwa komponen tren dan musiman dapat dimodelkan dengan baik dengan regresi. residual yang dihasilkan dari regresi kemudian ditambahkan dengan *noise* SSA untuk dimodelkan dengan DNN dengan prosedur yang sama sehingga diperoleh arsitektur DNN dalam Gambar 4.15



Gambar 4.15. Arsitektur DNN Terbaik dari SSA-DNN Yt_1_1

Berdasarkan model tersebut, diperoleh nilai *fits* dan *forecast* untuk 12 periode yang ditampilkan dalam Gambar 4.16



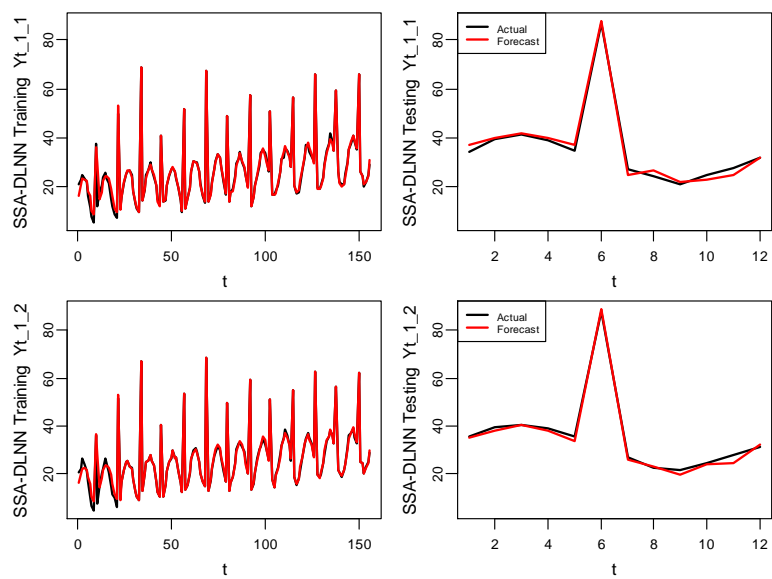
Gambar 4.16. Fits dan Forecast SSA-DNN Yt_1_1

Prosedur peramalan SSA-DNN yang sama diterapkan pada replikasi ke-2 hingga ke-5 sehingga diperoleh model DNN terbaik untuk *noise* SSA masing-masing replikasi seperti dalam Tabel 4.7

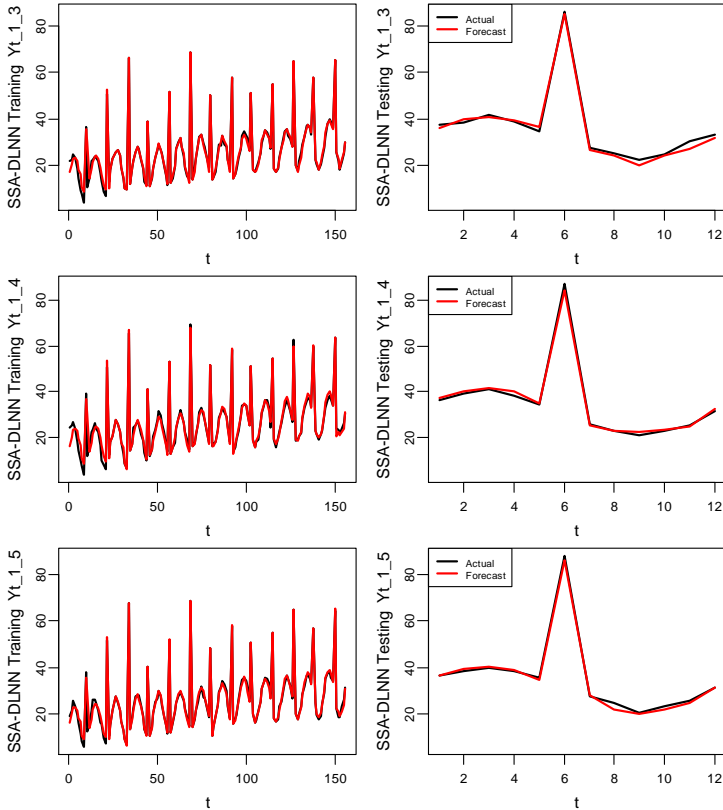
Tabel 4.7. Model DNN Hybrid Terbaik Skenario 1

<i>Residual</i>	<i>i</i>	<i>Neuron</i>		<i>k</i>	RMSE Testing
		<i>Hidden-1</i>	<i>Hidden-2</i>		
Yt_1_1	6	6	1	5	1,69
Yt_1_2	56	6	6	1	1,44
Yt_1_3	17	7	2	2	1,56
Yt_1_4	15	5	2	2	1,22
Yt_1_5	63	3	7	1	1,23

Berdasarkan model terbaik yang diperoleh untuk setiap replikasi dalam Tabel 4.7, diperoleh hasil *fits* dan *forecast* SSA-DNN dari skenario 1 replikasi 2 hingga 5 dalam Gambar 4.17



Gambar 4.17. *Fits dan Forecast* SSA-DNN Skenario 1



Gambar 4.17. (Lanjutan) *Fits* dan *Forecast* SSA-DNN Skenario 1

4.1.2 Skenario 2 dengan *Noise* mengikuti ESTAR(1)

Skenario 2 menggunakan pola *trend*, musiman dan variasi kalender yang sama seperti skenario 1, namun *noise* yang digunakan mengikuti model ESTAR (1). Seperti halnya pada skenario 1, pada skenario 2 data akan dimodelkan menggunakan metode ARIMAX, DNN-1, DNN-2, dan *hybrid* SSA-DNN dan dibandingkan berdasarkan RMSE dan sMAPE.

a. Peramalan Skenario 2 dengan ARIMAX

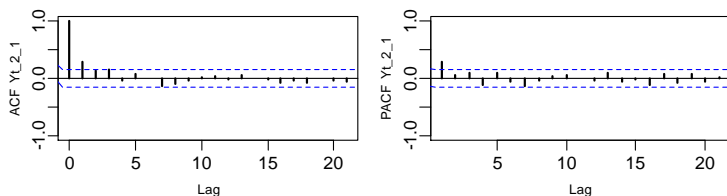
Time series regression dilakukan pada data simulasi skenario 2 sehingga menghasilkan residual yang akan dimodelkan

menggunakan ARIMA. Pada skenario 2 replikasi 1, hasil estimasi regresi *time series* ditunjukkan dalam Table 4.8

Tabel 4.8. TSR Yt_2_1

Prediktor	Estimasi	Std error	T	P
t	0,11	0,01	19,18	<0,001
$M_{1,t}$	19,05	0,98	19,53	<0,002
$M_{2,t}$	23,19	0,98	23,71	<0,003
$M_{3,t}$	24,40	0,98	24,88	<0,004
$M_{4,t}$	20,47	0,98	20,82	<0,005
$M_{5,t}$	18,78	0,99	19,05	<0,006
$M_{6,t}$	14,53	1,00	14,50	<0,007
$M_{7,t}$	8,15	1,04	7,85	<0,008
$M_{8,t}$	6,73	1,04	6,47	<0,009
$M_{9,t}$	3,78	1,06	3,55	<0,010
$M_{10,t}$	4,55	1,10	4,13	<0,011
$M_{11,t}$	9,23	1,05	8,79	<0,012
$M_{12,t}$	13,31	1,00	13,24	<0,013
$L_{1,t-1}$	57,64	2,37	24,37	<0,014
$L_{2,t-1}$	41,65	1,68	24,83	<0,015
$L_{3,t-1}$	34,96	1,93	18,09	<0,016
$L_{4,t-1}$	33,83	1,67	20,26	<0,017
$L_{1,t}$	0,46	2,36	0,19	0,847
$L_{2,t}$	-3,81	1,67	-2,28	0,024
$L_{3,t}$	1,43	1,94	0,74	0,464
$L_{4,t}$	0,81	1,67	0,49	0,627

Setelah regresi *time series* dilakukan, residual yang dihasilkan dimodelkan dengan ARIMA. Tahapan identifikasi dilakukan dengan bantuan ACF dan PACF dalam Gambar 4.18



Gambar 4.18. ACF dan PACF Residual TSR Yt_2_1

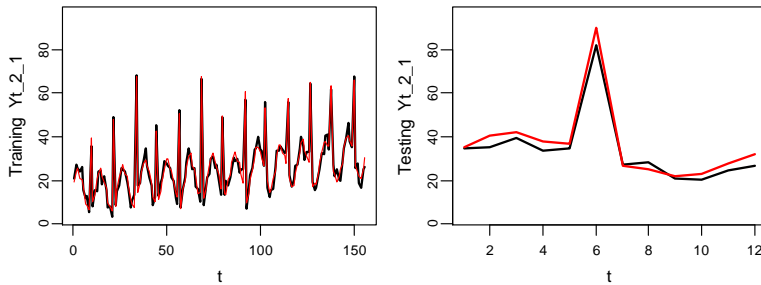
Gambar 4.18 menunjukkan ACF dengan pola *dies down* dan PACF dengan pola *cuts off* setelah lag-1. Sehingga kemungkinan

model ARIMA yang dapat dibentuk adalah ARIMA(1,0,0). Hasil Estimasi ARIMA di tujukan dalam Tabel 4.9.

Tabel 4.9. Estimasi Model ARIMA Residual TSR Yt_2_1

Parameter	Estimasi	SE	T	P
Intersep	-0.0061	0.318	0.019	0.508
AR1	0.2908	0.077	3.782	<0.001

Berdasarkan model tersebut, diperoleh nilai *fits* dan *forecast* untuk ARIMAX skenario 2 replikasi 1 pada 12 periode yang ditampilkan dalam Gambar 4.19



Gambar 4.19. Fits dan Forecast ARIMAX Yt_2_1

Gambar 4.19 menunjukkan bahwa pola *trend*, musiman dan variasi kalender pada data simulasi skenario 2 replikasi 1 masih dapat ditangkap dengan cukup baik oleh model ARIMAX walaupun data dibangkitkan menggunakan *noise* non-linier. Meskipun demikian, akurasi dari model ARIMAX ini perlu dibandingkan dengan metode DNN dan SSA-DNN untuk mengetahui perbandingan performa masing-masing model. Pemodelan ARIMAX dilakukan untuk replikasi ke-2 sampai ke-5 pada skenario 2 sehingga diperoleh model terbaik dalam persamaan:

- Persamaan ARIMAX skenario 2 replikasi 1

$$Y_{2.1,t} = (1 - 0,291Y_{t-1})^{-1} a_t + 0,111t + 19,051M_{1,t} + \dots + 0,814L_{4,t}$$

- Persamaan ARIMAX skenario 2 replikasi 2

$$Y_{2.2,t} = (1 - 0,331Y_{t-1})^{-1} a_t + 0,111t + 20,540M_{1,t} + \dots - 4,024L_{4,t}$$

- Persamaan ARIMAX skenario 2 replikasi 3

$$Y_{2,3,t} = (1 - 0,242Y_{t-1})^{-1}a_t + 0,099t + 21,028M_{1,t} + \dots + 0,302L_{4,t}$$

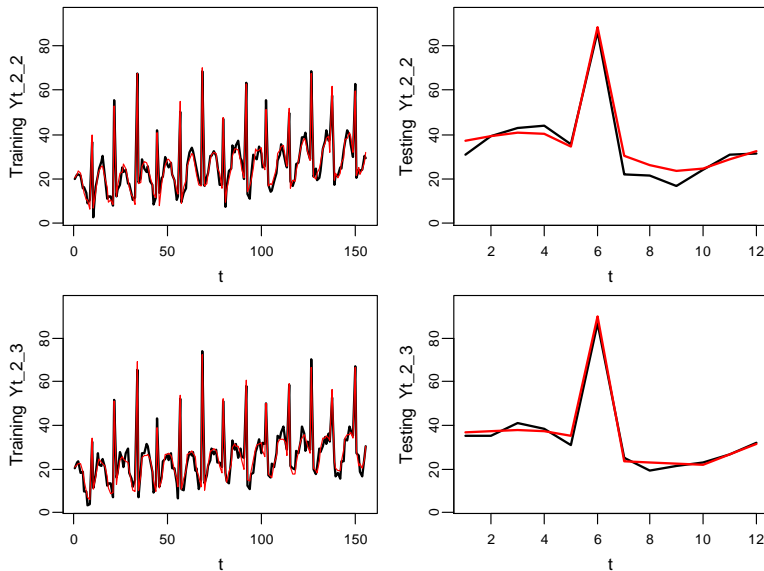
d. Persamaan ARIMAX skenario 2 replikasi 4

$$Y_{2,4,t} = (1 - 0,310Y_{t-1})^{-1}a_t + 0,113t + 17,937M_{1,t} + \dots + 3,513L_{4,t}$$

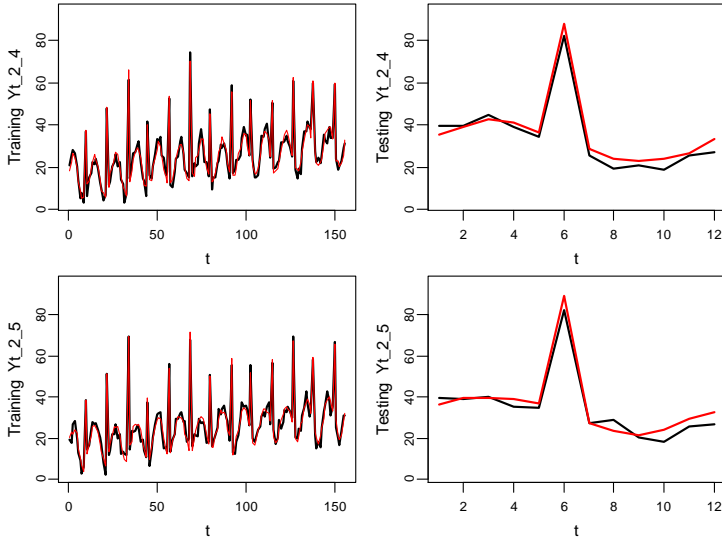
e. Persamaan ARIMAX skenario 2 replikasi 5

$$Y_{2,5,t} = (1 - 0,329Y_{t-1})^{-1}a_t + 0,103t + 20,176M_{1,t} + \dots - 1,230L_{4,t}$$

Pengujian *white noise* menggunakan Ljung-Box menunjukkan bahwa tidak semua model ARIMAX mampu memisahkan pola *noise* yang non-linier. Pada replikasi 4, residual yang dihasilkan tidak bersifat *white noise* sehingga kemungkinan terdapat pola musiman yang tidak tertangkap model sehingga masuk dalam *noise*. Berdasarkan model terbaik masing-masing replikasi, diperoleh hasil *fits* dan *forecast* ARIMAX 12 periode yang ditunjukkan dalam Gambar 4.20.



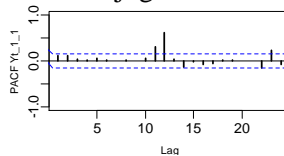
Gambar 4.20. *Fits dan Forecast ARIMAX Skenario 2*



Gambar 4.20 (Lanjutan) *Fits dan Forecast ARIMAX Skenario 2*

b. Peramalan Skenario 2 dengan DNN-1

Preprocessing DNN-1 dilakukan dengan menstandarisasi data. Tahap pertama dalam peramalan menggunakan DNN-1 adalah memilih variabel input. Pemilihan input dilakukan berdasarkan lag-PACF yang signifikan. Selain itu variabel *dummy* musiman dan variasi kalender juga ditambahkan sebagai input.

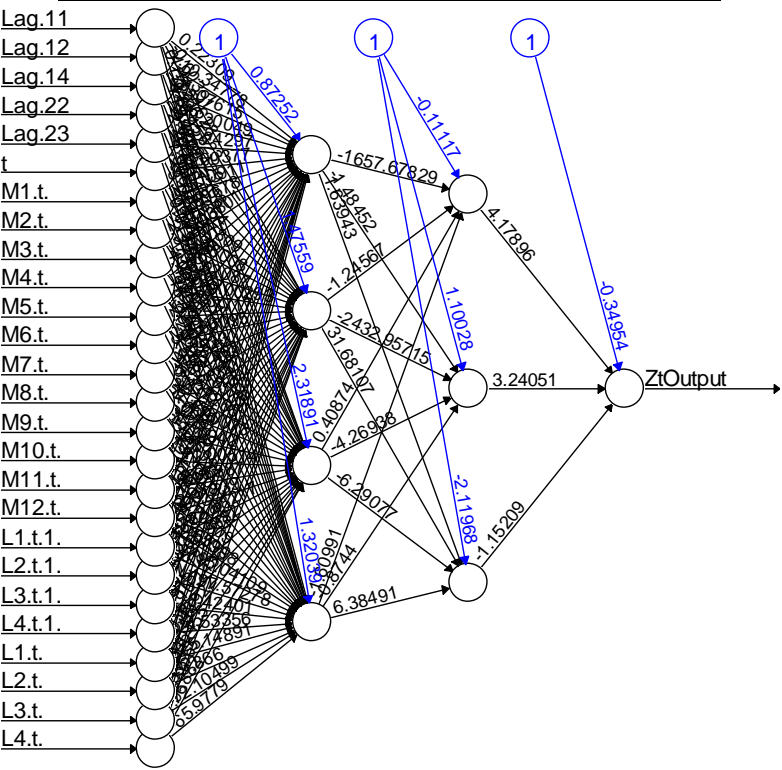


Gambar 4.21. PACF Y_{t-1}

Gambar 4.21 menunjukkan PACF lag-11, 12, 14, 22 dan lag-23 signifikan sehingga kelima lag-tersebut dijadikan input DNN-1 bersama variabel *dummy*. Pemilihan jumlah *neuron* pada setiap *hidden layer* dilakukan dengan *trial and error* yaitu mencoba sebanyak 1 sampai 10 *neuron* pada setiap *hidden*. Tabel 4.10 menunjukkan ringkasan dari 100 model DNN-1 untuk skenario 1 replikasi 1.

Tabel 4.10. Pemilihan Jumlah Neuron DNN-1 Yt_2_1

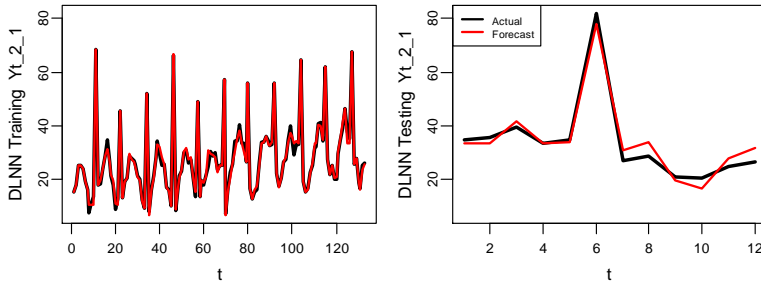
<i>i</i>	Neuron H-1	Neuron H-2	<i>k</i>	RMSE	
				Training	Testing
1	1	1	5	2,80	3,77
2	2	1	3	2,83	5,49
...
24	4	3	4	1,29	3,15
...
100	10	10	3	0,08	10,99



Gambar 4.22. Arsitektur DNN Yt_2_1

Tabel 4.10 dan Gambar 4.22 menunjukkan bahwa 4 *neuron* pada *hidden layer* pertama dan 3 *neuron* pada *hidden layer* ke-2 merupakan kombinasi *neuron* optimum yang menghasilkan RMSE

testing terkecil. Model terbaik ini diperoleh pada percobaan *set seet* ke-4. Berdasarkan model terbaik DNN-1 tersebut diperoleh nilai *fits* dan *forecast* untuk skenario 2 replikasi 1 yang ditampilkan pada Gambar 4.23.



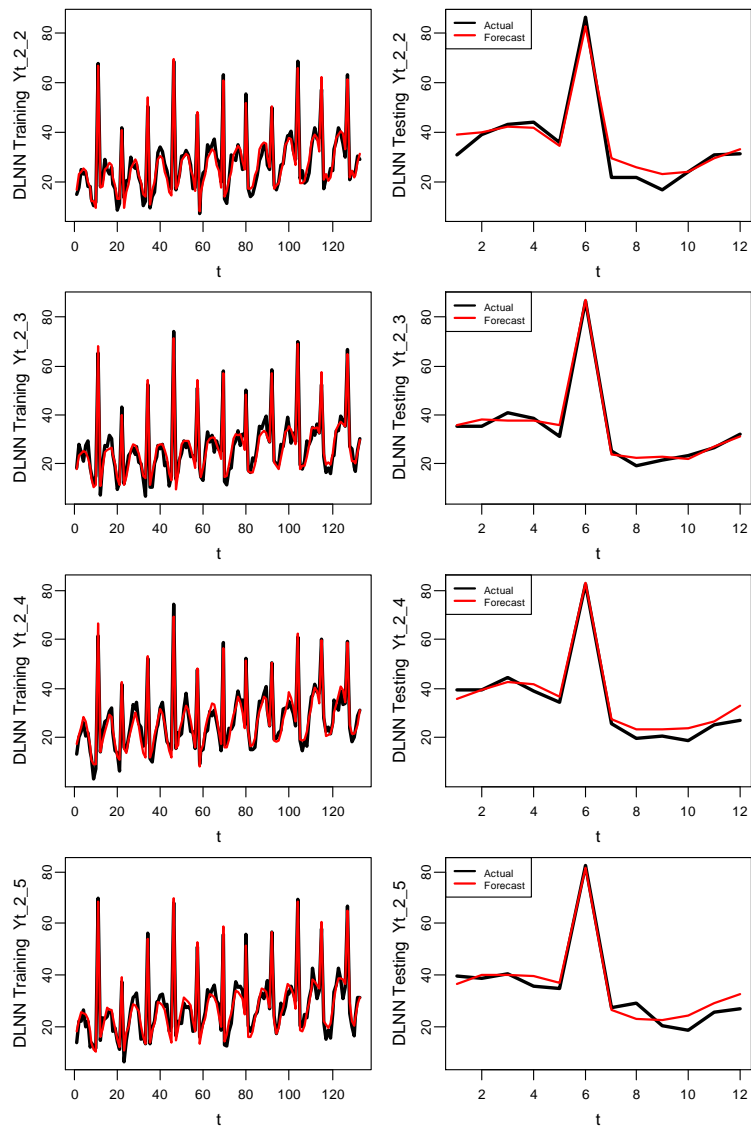
Gambar 4.23. *Fits dan Forecast DNN-1 Yt_2_1*

Gambar 4.23 menunjukkan bahwa data yang mengandung *noise* non-linier dapat ditangkap dengan baik oleh model DNN-1 yang menggunakan input lag PACF data asli. Peramalan menggunakan DNN-1 untuk replikasi 2 sampai 5 dilakukan dengan cara yang sama dengan replikasi 1. Tabel 4.11 menunjukkan ringkasan informasi model DNN-1 terbaik untuk replikasi 1 hingga replikasi 5.

Tabel 4.11. Model DNN-1 Terbaik Skenario 2

i	<i>Neuron</i>		k	RMSE <i>Testing</i>
	<i>Hidden-1</i>	<i>Hidden-2</i>		
24	4	3	4	3,15
35	5	4	4	4,18
21	1	3	2	2,15
72	2	8	5	3,16
72	2	8	4	3,59

Berdasarkan model terbaik untuk setiap replikasi yang telah diperoleh diperoleh kemudian dapat dihitung nilai *fits* dan *forecast* dari setiap replikasi pada skenario 2 yang ditampilkan dalam seperti dalam Gambar 4.24

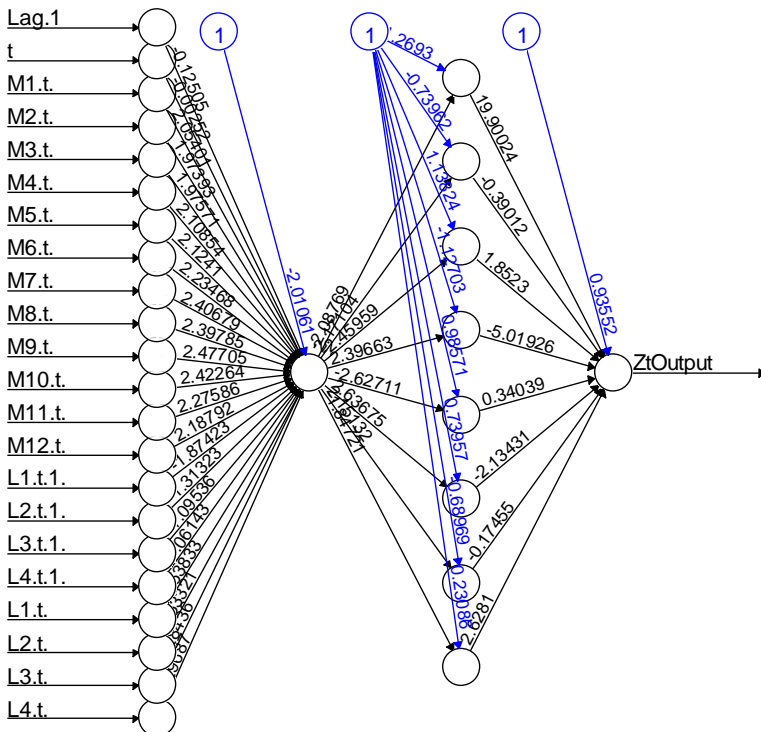


Gambar 4.24. *Fits dan Forecast DNN-1 Skenario 2*

Gambar 4.24 menunjukkan bahwa metode DNN-1 secara konsisten mampu memodelkan dan meramal data yang mengandung pola *noise* non-linier.

c. Peramalan Skenario 2 dengan DNN-2

Pada peramalan dengan DNN-2 *preprocessing* dilakukan dengan cara yang sama seperti DNN-1. Input yang digunakan adalah lag-1 sesuai model ARIMAX. Model terbaik untuk replikasi pertama yaitu model dengan 1 neuron pada *hidden layer* pertama dan 7 neuron pada *hidden layer* ke-2. Arsitektur DNN-2 terbaik yang diperoleh ditampilkan sebagai berikut



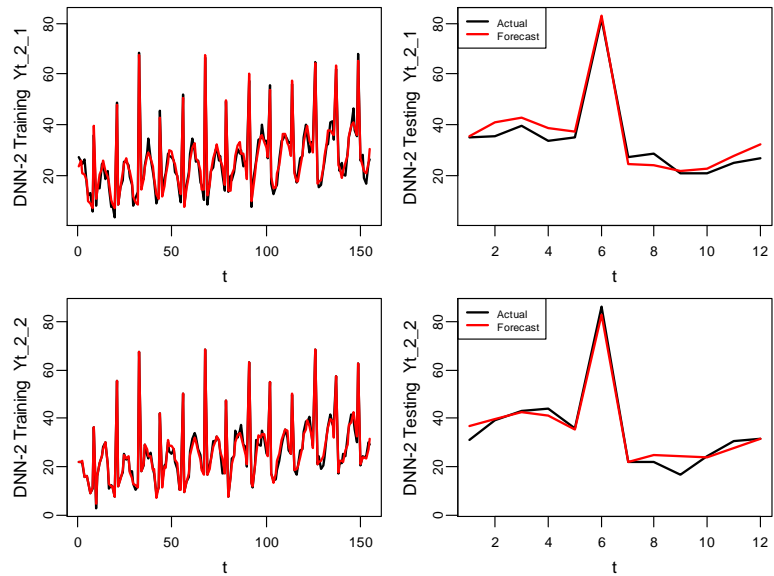
Gambar 4.25. Arsitektur DNN-2 Terbaik Yt_2_1

Langkah yang sama dilakukan terhadap replikasi 2 hingga ke-5 sehingga dihasilkan ringkasan informasi model DNN-2 terbaik untuk setiap replikasi skenario 2 yang ditampilkan dalam Tabel 4.12

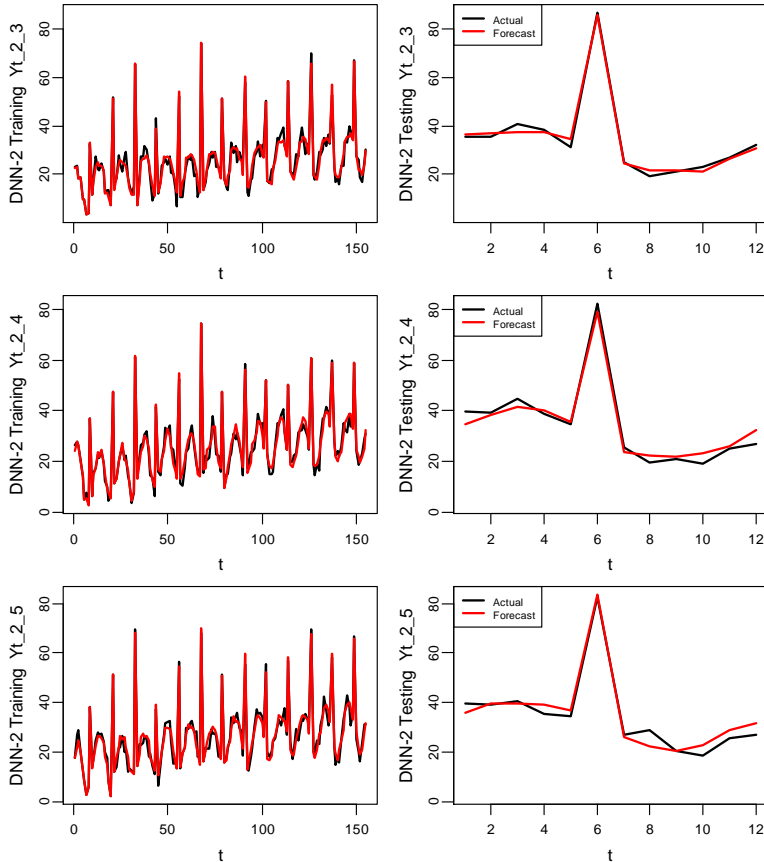
Tabel 4.12. Model DNN-2 Terbaik Skenario 2

<i>i</i>	<i>Neuron</i>		<i>k</i>	RMSE <i>Testing</i>
	<i>Hidden-1</i>	<i>Hidden-2</i>		
71	1	7	1	3,49
25	5	3	2	3,36
72	2	8	4	1,97
32	2	4	3	3,00
13	3	2	2	3,24

Berdasarkan model DNN-2 terbaik yang diperoleh dari setiap replikasi data, hasil pemodelan berupa *fits* dan *forecast* ditampilkan sebagai berikut.



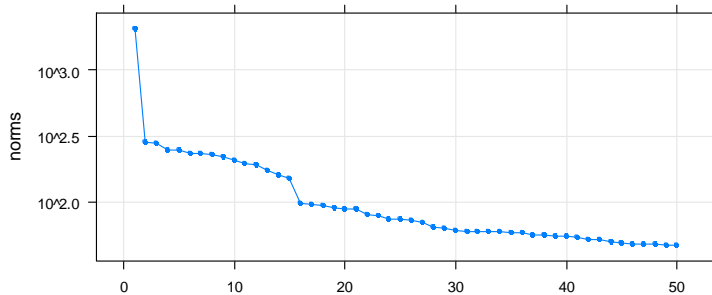
Gambar 4.26. Fits dan Forecast DNN-2 Skenario 2



Gambar 4.26 (Lanjutan) Fits dan Forecast DNN-2 Skenario 2

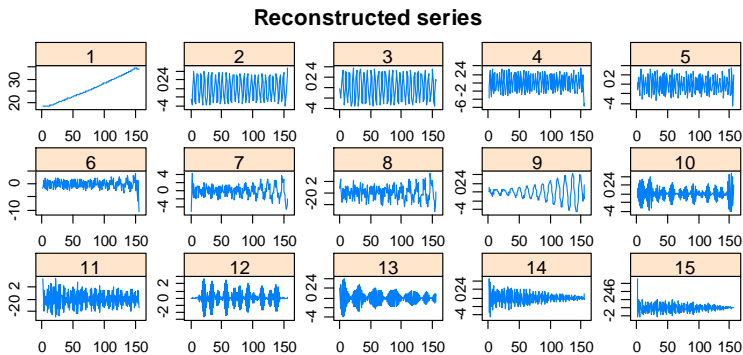
d. Peramalan Skenario 2 dengan SSA-DNN

Pemodelan menggunakan model *hybrid* SSA-DNN untuk skenario 2 yang mengandung pola non-linier dilakukan dengan prosedur yang sama seperti SSA-DNN pada skenario 1. Identifikasi pola tren dan musiman dilakukan melalui plot *eigenvalues* yang ditampilkan dalam Gambar 4.27. Tahap ini bertujuan untuk memisahkan komponen *noise* dari pola tren dan musiman data asli



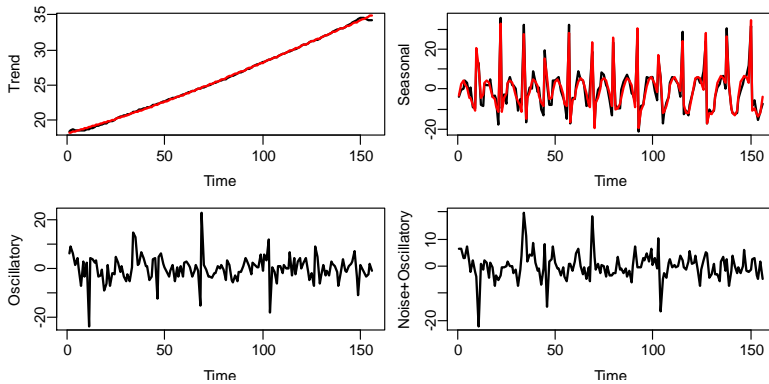
Gambar 4.27. Plot Eigenvalues Yt_2_1

Berdasarkan Gambar 4.27 dapat dilihat bahwa penurunan tajam terjadi pada *eigenvalue* ke-15 menuju 16 sehingga 16 *eigen triples* pertama dikelompokkan sebagai komponen *trend* dan musiman sedangkan *eigenvalues* ke-16 dan seterusnya dikelompokkan sebagai *noise*. Rekonstruksi individu dilakukan untuk mengetahui pola *trend* dan musiman seperti pada Gambar 4.28.



Gambar 4.28. Rekonstruksi Individu Yt_2_1

Gambar 4.28 menunjukkan bahwa *eigen triples* pertama memiliki pola *trend* sedangkan *eigen triples* ke-2 hingga ke-15 memiliki pola musiman. Selanjutnya rekonstruksi agregat dilakukan untuk membentuk komponen *trend* dan komponen musiman untuk dimodelkan menggunakan regresi polinomial dan regresi *dummy* dengan hasil pada Gambar 4.29

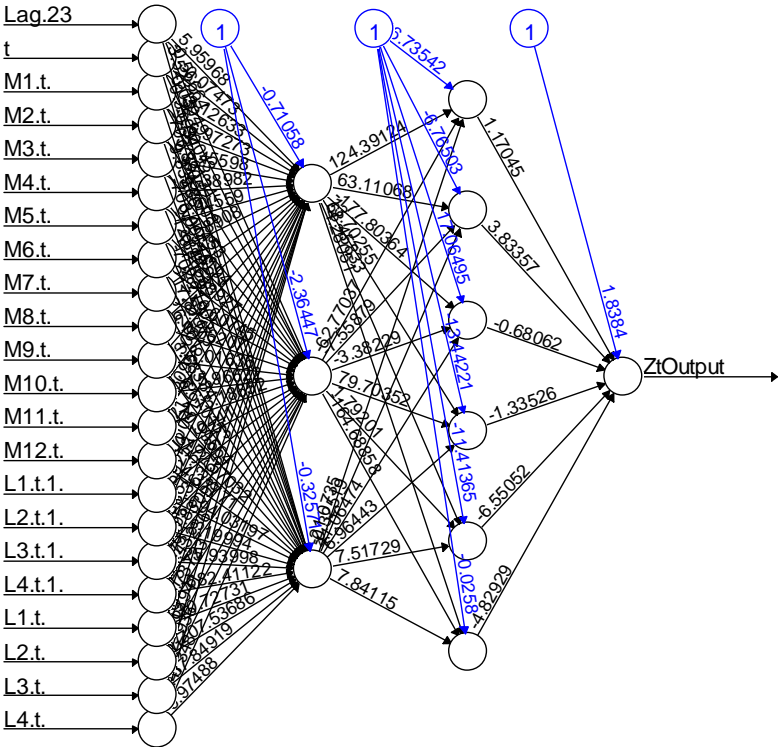


Gambar 4.29. Rekonstruksi & Pemodelan Agregat Yt_2_1

Gambar 4.29 menunjukkan bahwa pola *trend* dan musiman mampu ditangkap oleh model regresi. Residual dari kedua model regresi ini ditambahkan dengan komponen noise untuk dimodelkan menggunakan DNN.

Prosedur DNN yang dilakukan sama dengan prosedur yang diterapkan pada SSA-DNN skenario 1. Input PACF diidentifikasi melalui plot PACF yang menunjukkan bahwa lag-23 adalah satu-satunya lag yang signifikan. Sehingga lag-23 dan variabel *dummy* digunakan sebagai input DNN. Jumlah neuron yang menghasilkan RMSE *testing* optimum adalah 3 *neuron hidden layer-1* dan 4 *neuron hidden layer-1* pada percobaan *set seed* ke-5. Arsitektur model terbaik DNN yang dihasilkan ditampilkan dalam Gambar 4.30.

Lag-23 menjadi satu-satunya lag yang signifikan pada *noise* SSA pecahan *outflow* Rp 100.000,-. Hal ini menunjukkan bahwa SSA dapat memisahkan komponen musiman dengan periode 12 dengan baik sehingga lag 12 yang merupakan faktor musiman bulan Masehi yang umumnya terjadi dapat terekonstruksi sebagai musiman (tidak termasuk dalam *noise*). Namun lag-23 muncul dimungkinkan karena adanya efek variasi kalender yang tidak tertangkap SSA.



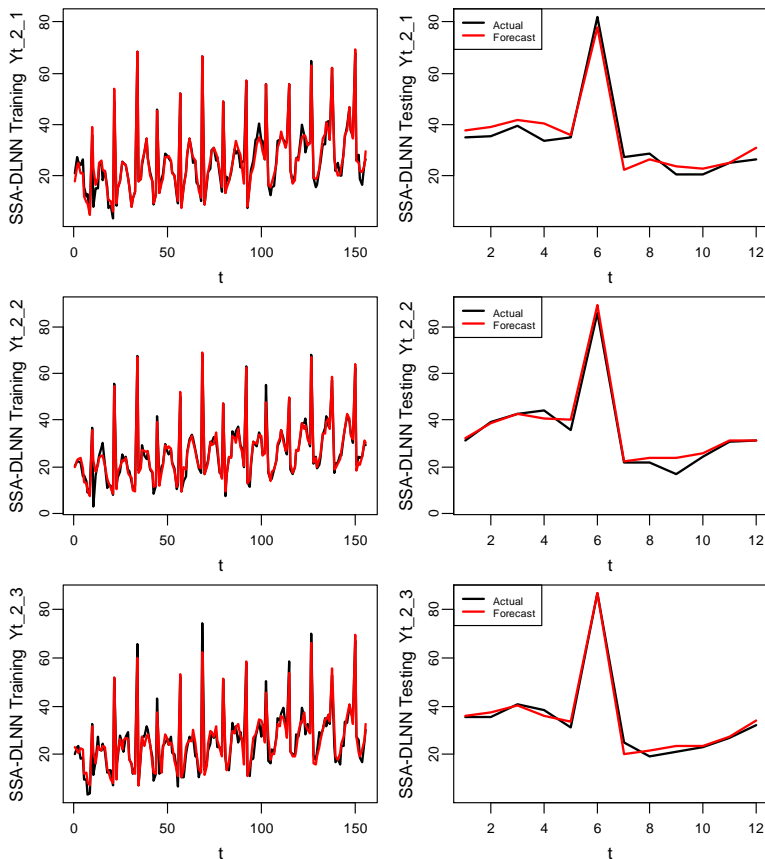
Gambar 4.30. Arsitektur DNN Terbaik dari SSA-DNN Yt_2_1

Pemodelan dengan SSA-DNN dilanjutkan untuk replikasi ke-2 hingga replikasi ke-5 dengan prosedur yang sama. Kombinasi *neuron* optimum ditampilkan dalam Tabel 4.13 sebagai model DNN terbaik dalam memodelkan residual dari SSA.

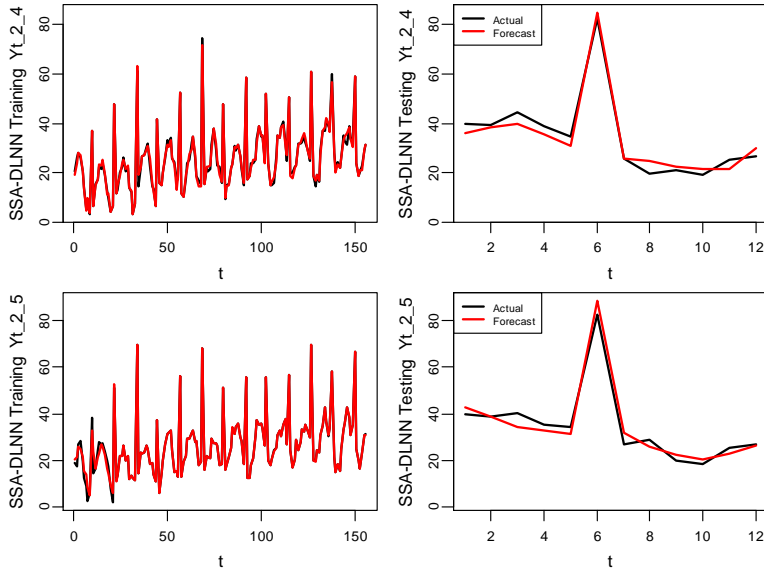
Tabel 4.13. Model DNN Hybrid Terbaik Skenario 2

Residual	<i>i</i>	<i>Neuron</i>		<i>k</i>	RMSE Testing
		<i>Hidden-1</i>	<i>Hidden-2</i>		
Yt_2_1	53	3	6	5	3,57
Yt_2_2	73	3	8	4	2,92
Yt_2_3	2	2	1	4	2,15
Yt_2_4	4	4	1	5	3,29
Yt_2_5	100	10	10	2	3,51

Berdasarkan model terbaik DNN dalam Tabel 4.13 diperoleh *fits* dan *forecast* untuk *noise* SSA untuk setiap replikasi. *Fits* dan *forecast* ini selanjutnya ditambahkan dengan hasil *fits* dan *forecast* dari komponen *trend* dan musiman dari model regresi setiap replikasi. Sehingga diperoleh *fits* dan *forecast* untuk SSA-DNN skenario 2 replikasi 1 sampai replikasi 5 seperti yang ditampilkan dalam Gambar 4.31



Gambar 4.31. *Fits* dan *Forecast* SSA-DNN Skenario 2

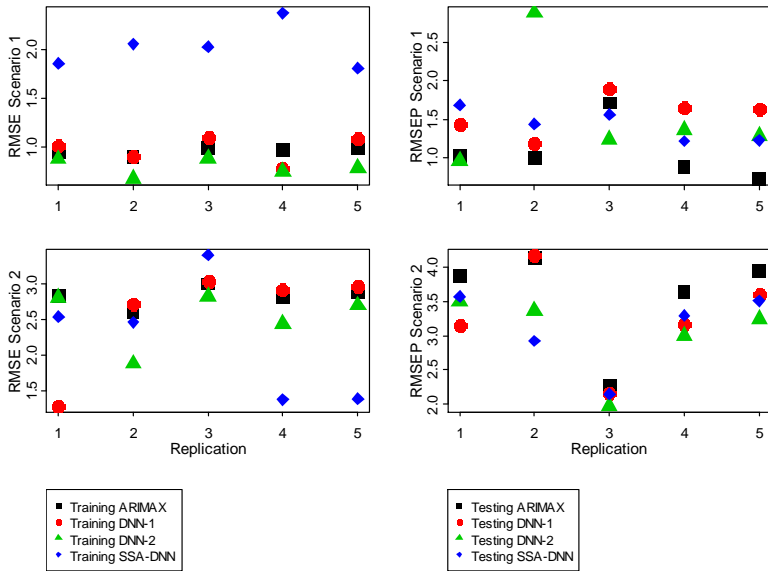


Gambar 4.31 (Lanjutan) *Fits dan Forecast SSA-DNN Skenario 2*

Gambar 4.31 menunjukkan bahwa SSA-DNN mampu mendekomposisi pola deret waktu secara optimal pada data simulasi karena pola data simulasi yang konsisten. Sehingga output ramalan yang dihasilkan juga dapat mengikuti pola data *testing* skenario 2 yang mengandung pola non-linier dengan baik. Simulasi ini juga membuktikan bahwa model *hybrid* SSA-DNN mampu menangkap efek variasi kalender.

4.1.3 Perbandingan Performa ARIMAX, DNN dan SSA-DNN pada Data Simulasi

Perbandingan performa ARIMAX, DNN dan SSA_DNN dievaluasi berdasarkan nilai RMSE dan sMAPE baik pada data *training* maupun data *testing*. Ringkasan akurasi dari keempat model untuk skenario 1 dan skenario 2 pada replikasi 1 hingga replikasi 5 ditampilkan dalam Gambar 4.32

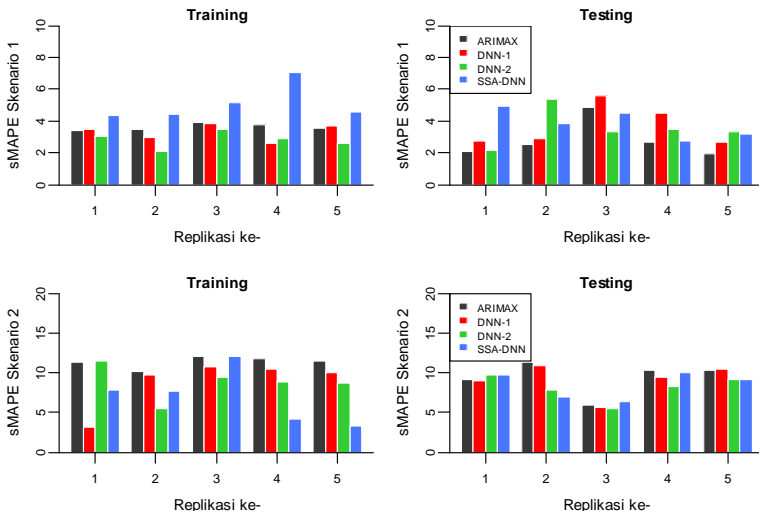


Gambar 4.32. Perbandingan RMSE Training & Testing Simulasi

Gambar 4.32 menunjukkan bahwa pada skenario 1 yang mengandung pola non-linier, ARIMAX unggul pada 4 replikasi dari 5 replikasi pada data *testing*, namun DNN-2 merupakan yang terbaik pada *training* semua replikasi. Metode SSA-DNN memiliki akurasi paling rendah pada *training* namun cukup bagus dalam *testing*. DNN-1 dan DNN-2 bagus di *training* karena memang merupakan bagian dari *machine learning* yang kuat pada data *training*. Sedangkan ARIMAX sangat bagus di *training* karena data dibangkitkan menggunakan persamaan model ARIMAX. Namun perbedaan akurasi antar metode pada skenario 1 tidak terlalu jauh berbeda. Hal ini menunjukkan bahwa memang pada penerapannya, metode DNN dapat digunakan pada data dengan pola non-linier maupun linier.

Pada data skenario 2 yang mengandung pola non-linier, ARIMAX merupakan model dengan akurasi paling rendah

dibandingkan metode lain. Secara umum, pada skenario 2 performansi DNN-2 dan SSA-DNN lebih bagus dibandingkan DNN-1 dan juga ARIMAX. Perbandingan berdasarkan kriteria selain RMSE yaitu sMAPE ditampilkan dalam Gambar 4.33 sebagai pembandingan tambahan.



Gambar 4.33. Perbandingan sMAPE Training & Testing Simulasi Simulasi

Perbandingan kriteria sMAPE menunjukkan hal yang serupa dengan RMSE. Pada skenario 1 data *training*, DNN-2 paling dominan, sedangkan pada data *testing* model terbaik didominasi oleh ARIMAX. Sementara itu, pada skenario 2 SSA-DNN dan DNN-2 sama baiknya pada *training* maupun *testing*. Sedangkan ARIMAX tidak lebih baik dibandingkan metode lain pada hampir semua replikasi. Rasio akurasi DNN-1, DNN-2 dan SSA-DNN terhadap ARIMAX ditampilkan dalam Tabel 14 dan Tabel 15 untuk mempermudah pembandingan.

Tabel 4.14. Rasio Akurasi Data *Training* DNN dan SSA-DNN pada Data Simulasi terhadap ARIMAX

Data	RMSE			sMAPE		
	DNN-1	DNN-2	SSA-DNN	DNN-1	DNN-2	SSA-DNN
Yt_1_1	1.06	0.92	1.95	1.02	0.89	1.27
Yt_1_2	1.00	0.76	2.28	0.86	0.60	1.29
Yt_1_3	1.10	0.89	2.03	0.99	0.90	1.33
Yt_1_4	0.81	0.78	2.43	0.70	0.78	1.88
Yt_1_5	1.09	0.79	1.82	1.05	0.73	1.30
Yt_2_1	0.46	1.00	0.90	0.28	1.01	0.68
Yt_2_2	1.05	0.72	0.94	0.96	0.54	0.75
Yt_2_3	1.01	0.94	1.14	0.90	0.78	1.01
Yt_2_4	1.04	0.87	0.49	0.88	0.75	0.35
Yt_2_5	1.03	0.94	0.48	0.88	0.76	0.29

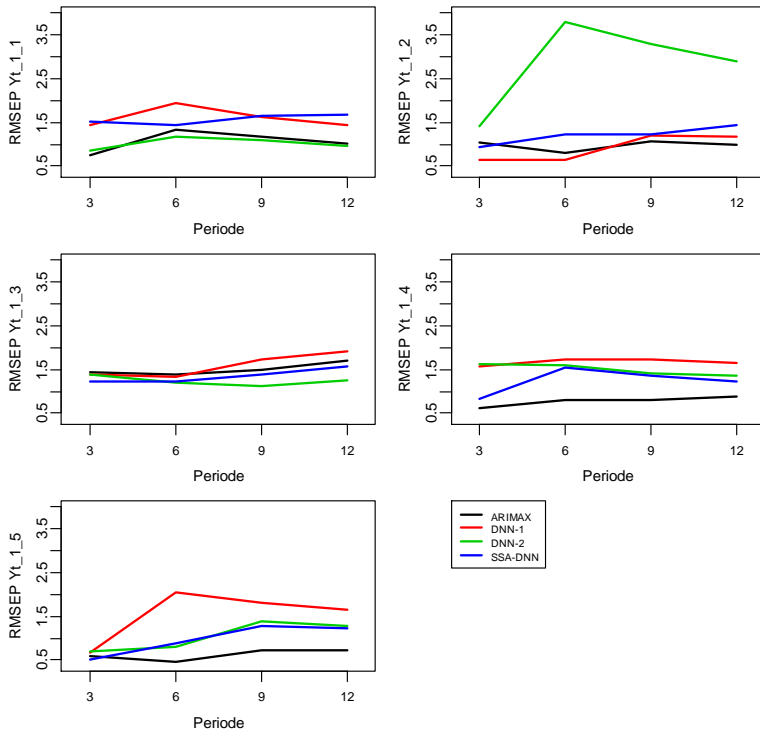
sedangkan rasio akurasi data *testing* ditunjukkan dalam Tabel 4.15

Tabel 4.15. Rasio Akurasi Data *Testing* DNN dan SSA-DNN pada Data Simulasi terhadap ARIMAX

Data	RMSEP			sMAPEP		
	DNN-1	DNN-2	SSA-DNN	DNN-1	DNN-2	SSA-DNN
Yt_1_1	1.39	0.93	1.64	1.31	1.04	2.32
Yt_1_2	1.20	2.92	1.45	1.16	2.13	1.52
Yt_1_3	1.11	0.73	0.91	1.16	0.69	0.93
Yt_1_4	1.88	1.55	1.38	1.67	1.29	1.01
Yt_1_5	2.25	1.76	1.68	1.38	1.70	1.63
Yt_2_1	0.81	0.90	0.92	0.99	1.07	1.07
Yt_2_2	1.01	0.82	0.71	0.97	0.69	0.62
Yt_2_3	0.95	0.87	0.94	0.95	0.91	1.06
Yt_2_4	0.87	0.82	0.90	0.91	0.80	0.96
Yt_2_5	0.91	0.82	0.89	1.02	0.88	0.89

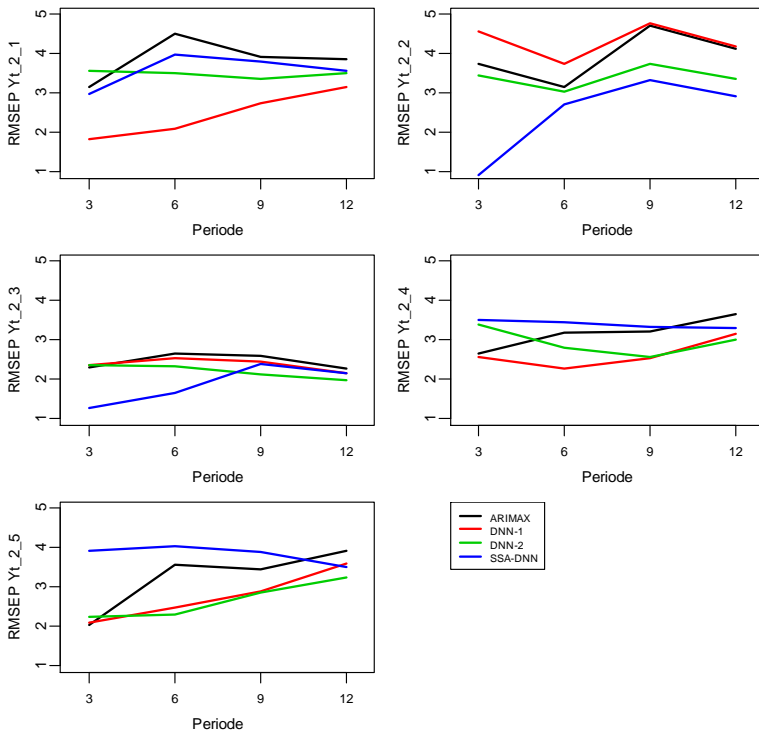
Rasio RMSE menunjukkan bahwa pada skenario 1 ARIMAX adalah model terbaik. Rasio lebih dari 1 pada 3 dari 5 replikasi menurut RMSEP dan 4 dari 5 replikasi menurut sMAPEP. Pada skenario 2, 3 dari 5 replikasi menurut RMSEP dan 4 dari 5 replikasi menurut sMAPE menunjukkan DNN-2 adalah model terbaik. Selain perbandingan akurasi pada *training* dan data *testing* sepanjang 12 periode, evaluasi juga dilakukan pada kebaikan jangka ramalan

yaitu pada ramalan 3 periode, 6 periode, 9 periode dan 12 periode yang ditampilkan dalam Gambar 4.34 dan Gambar 4.35.



Gambar 4.34. Perbandingan RMSEP Skenario 1 Menurut Panjang Periode

Pada skenario 1, ramalan jangka pendek 3 periode yang terbaik dihasilkan oleh model ARIMAX dan SSA-DNN. Sementara untuk ramalan 12 periode, model terbaik adalah model ARIMAX yang unggul pada 3 dari 5 replikasi. Secara umum pada skenario 1 ARIMAX memiliki konsistensi akurasi yang paling stabil. Akurasi antar metode lebih homogen pada periode pendek dan cenderung semakin beragam pada periode yang lebih panjang. Metode DNN-2 juga menunjukkan performa yang sangat baik pada skenario 1 kecuali pada replikasi 2.



Gambar 4.35. Perbandingan RMSEP Skenario 2 Menurut Panjang Periode

Pada skenario 2, ramalan jangka pendek 3 periode yang terbaik dihasilkan oleh model DNN-1 dan SSA-DNN. Sementara untuk ramalan 12 periode, model terbaik adalah model DNN-2. Metode DNN-2 memiliki konsistensi akurasi yang paling stabil dari 3 periode hingga 12 periode. Akurasi antar metode lebih beragam pada periode pendek dan cenderung semakin homogen pada periode yang lebih panjang

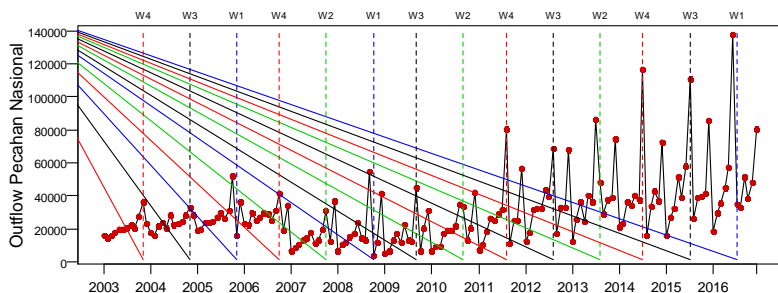
4.2 Karakteristik *Inflow* dan *Outflow* di Indonesia

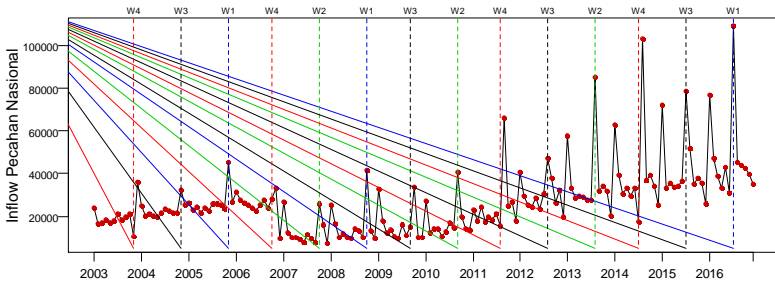
Karakteristik *inflow* dan *outflow* setiap pecahan secara deskriptif dijelaskan dalam Tabel 4.16

Tabel 4.16. Statistika Deskriptif *inflow* dan *outflow*

Data	Pecahan	Mean	Standar Deviasi	Median	Skewness	Kurtosis
<i>Outflow</i>	100000	14650,00	12827,00	10452,00	2,28	6,80
	50000	13013,00	7027,00	12172,00	1,32	2,86
	20000	969,40	751,80	743,40	1,84	5,29
	10000	624,20	624,40	492,40	3,83	16,69
	5000	464,60	592,20	319,90	4,07	18,72
	2000	148,00	312,40	49,20	3,70	14,58
	1000	91,08	100,67	59,71	1,76	3,81
<i>Inflow</i>	100000	12549,00	10836,00	9164,00	2,49	8,05
	50000	12382,00	5725,00	12316,00	1,34	3,49
	20000	946,20	486,10	719,40	1,19	0,76
	10000	573,30	252,00	531,30	2,49	9,20
	5000	418,00	204,00	370,50	2,66	9,79
	2000	107,20	131,10	14,00	0,95	-0,10
	1000	95,00	70,11	82,70	0,73	-0,23

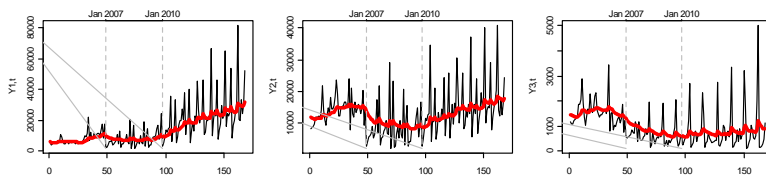
Tabel 4.16 menunjukkan bahwa rata-rata dan standar deviasi berbanding lurus terhadap besarnya nominal pecahan. Nilai skewness lebih dari nol menunjukkan bahwa terdapat kemiringan pada sebaran data *inflow* maupun *outflow*. Hampir semua pecahan memiliki kemiringan sebaran data kearah kanan, hal ini dimungkinkan terjadi karena terdapat beberapa data yang jauh lebih besar dibandingkan data lainnya atau *outlier*. Melonjaknya *inflow* dan *outflow* yang menyebabkan kemiringan sebaran data juga diduga karena adanya pengaruh Hari Raya Idul Fitri. Perkembangan *inflow* dan *outflow* secara nasional ditampilkan dalam Gambar 4.36 dan 4.37

**Gambar 4.36.** Perkembangan *Outflow* Nasional

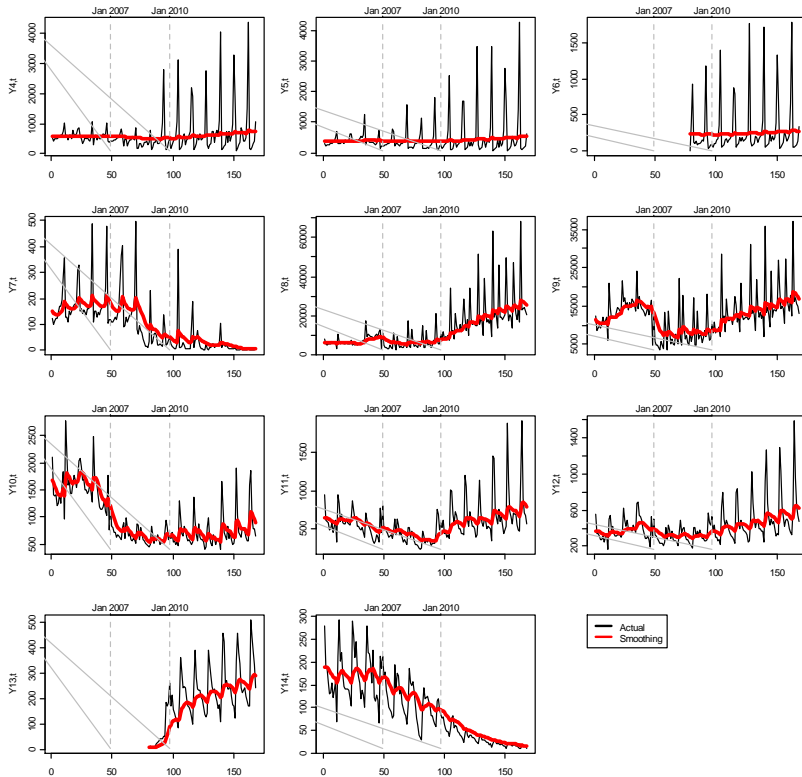


Gambar 4.37. Perkembangan *Inflow* Nasional

Gambar 4.36 dan Gambar 4.37 yang dikutip dari Saputri (2017) menunjukkan bahwa *outflow* dan *inflow* mengalami peningkatan tinggi bertepatan pada terjadinya Hari Raya Idul Fitri dan 1 bulan sebelum serta setelahnya. Selain itu, berdasarkan kedua plot, *inflow* dan *outflow* mengalami perubahan *trend* yang berbeda dan cenderung lebih rendah pada tahun 2007 hingga akhir 2010 dibandingkan tahun tahun lainnya. Penurunan ini bertepatan dengan diberlakukannya kebijakan Bank Indonesia yang membatasi setoran tunai ke Bank Indonesia untuk uang kertas yang tidak layak. Sehingga mempengaruhi *inflow* yang cenderung turun, dan secara tidak langsung akan mempengaruhi *outflow* karena terbatasnya arus *inflow*. Sehingga dalam penelitian ini akan digunakan *dummy* yang membedakan antara *inflow* dan *outflow* sebelum tahun 2007, antara 2007 hingga akhir 2010 serta tahun 2011 dan seterusnya. Gambaran perubahan pola ini didukung oleh hasil *smoothing* setiap pecahan yang ditampilkan sebagai berikut

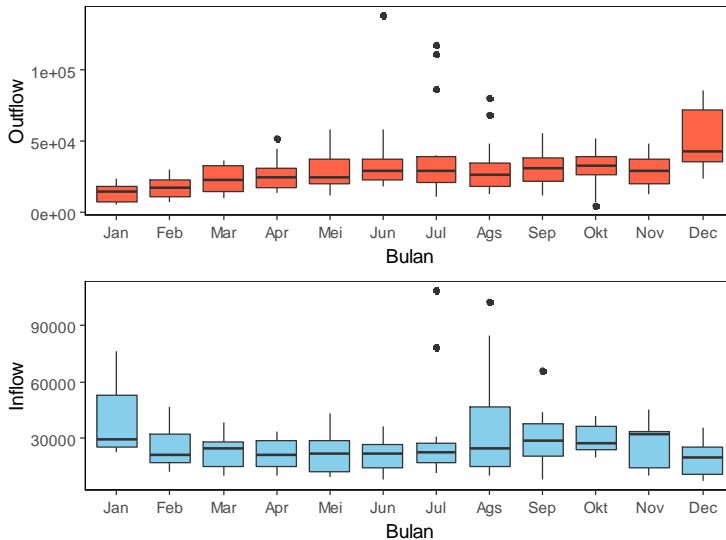


Gambar 4.38. Aktual & Hasil Smoothing Data Outflow & Inflow



Gambar 4.38 (Lanjutan) Actual & Hasil Smoothing Data Outflow & Inflow

Gambar 4.38 mendukung dugaan bahwa kebijakan pembatasan *inflow* memberikan pengaruh pada tren beberapa pecahan. Selain identifikasi tren, pola musiman juga perlu diidentifikasi. Pola musiman dapat teridentifikasi secara sekilas dalam Gambar 4.37 dan Gambar 4.38 dari adanya perulangan pola naik dan turun yang terusterjadi dari awal data. Pola musiman dapat diidentifikasi secara visual dengan lebih baik menggunakan *boxplot* yang ditampilkan dalam Gambar 4.39



Gambar 4.39. Boxplot Outflow & Inflow Menurut Bulan

Gambar 4.39 menunjukkan *outflow* cenderung meningkat menuju akhir tahun. Hal ini kemungkinan terjadi karena meningkatnya konsumsi di masyarakat untuk merayakan Natal dan Tahun Baru. Sedangkan *outlier* banyak terjadi pada bulan Juli & Agustus, hal ini bertepatan dengan terjadinya Idul Fitri dan juga terjadi pada *inflow*. Sedangkan pola musiman *inflow* cenderung tinggi di awal tahun dan bulan November.

4.3 Peramalan *Inflow* dan *Outflow* Menggunakan ARIMAX

Pemodelan data *inflow* dan *outflow* dengan ARIMAX dilakukan dengan cara memodelkan data menggunakan *time series regression* dengan t , variabel *dummy* kebijakan, variabel *dummy* interaksi t dengan kebijakan, variabel *dummy* bulan dan *dummy* variasi kalender. Apabila residual *time series regression* telah memenuhi asumsi *white noise*, maka pemodelan residual menggunakan ARIMA tidak perlu dilakukan dan model *time series regression* dipilih sebagai metode terbaik. Sedangkan apabila

residual tidak bersifat *white noise* maka dimodelkan menggunakan ARIMA sehingga terbentuk model ARIMAX dengan residual *white noise*. Hasil estimasi parameter *time series regression* pada data *outflow* pecahan Rp.100.000,- ditampilkan dalam Tabel 4.17.

Tabel 4.17. Estimasi TSR *Outflow* Rp 100.000,-

Variabel	Estimasi	std,error	T	P-value
t	280,6	42,6	6,58	<0,0001
D ₁	18941,9	5659,2	3,35	0,0011
D ₂	12308,4	7282,1	1,69	0,0934
tD_1	-147,6	66,3	-2,23	0,0277
tD_2	-147,3	74,9	-1,97	0,0514
M _{1,t}	-17528,9	6167,3	-2,84	0,0052
M _{2,t}	-16331,7	6188,9	-2,64	0,0094
M _{3,t}	-18424,9	6070,5	-3,04	0,0029
M _{4,t}	-15692,5	6102,4	-2,57	0,0113
M _{5,t}	-15242,3	6060,1	-2,52	0,0131
M _{6,t}	-12726,1	6098,2	-2,09	0,0389
M _{7,t}	-11210,6	6141,3	-1,83	0,0703
M _{8,t}	-16193,9	6129,6	-2,64	0,0093
M _{9,t}	-12917,8	6166,9	-2,09	0,0382
M _{10,t}	-15518,6	6262,9	-2,48	0,0145
M _{11,t}	-23686,0	6353,4	-3,73	0,0003
M _{12,t}	-8195,5	6190,6	-1,32	0,1879
L _{1,t-1}	18198,0	2611,9	6,97	<0,0001
L _{2,t-1}	3354,6	2037,9	1,65	0,1022
L _{3,t-1}	-4043,8	2157,8	-1,87	0,0632
L _{4,t-1}	1272,7	1736,2	0,73	0,4649
L _{1,t}	2815,7	2155,8	1,31	0,1939
L _{2,t}	11505,2	1843,1	6,2	<0,0001
L _{3,t}	16131,4	2444,7	6,6	<0,0001
L _{4,t}	17060,8	2106,8	8,1	<0,0001

Residual dari *time series regression* ini selanjutnya akan dimodelkan menggunakan ARIMA dengan orde yang diduga dari plot ACF dan PACF. Namun jika residual telah bersifat *white noise*, maka pemodelan residual menggunakan ARIMA tidak perlu dilakukan. Pengujian *white noise* dilakukan menggunakan uji Ljung-Box dengan hasil pada Tabel 4.18.

Tabel 4.18. Ljung-Box Residual TSR *Outflow* Rp 100.000,-

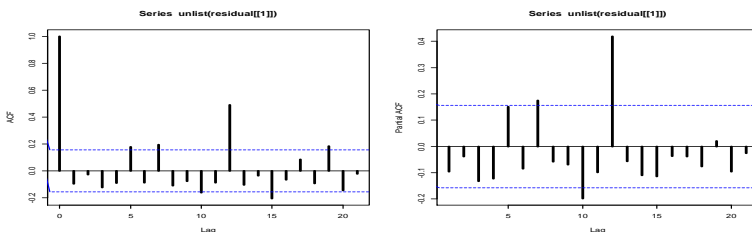
Lag	Chi-square	P-value
6	11,465	0,075
12	62,741	0,000
18	74,419	0,000
24	100,311	0,000

Tabel 4.18 menunjukan bahwa sampai *lag* ke-6 residual *time series* masih bersifat *white noise*, namun hingga *lag* ke-12 dan seterusnya residual tidak bersifat *white noise* sehingga perlu dimodelkan menggunakan ARIMA. Orde *d* dari ARIMA bernilai tidak nol jika residual belum stasioner dalam *mean*. Stasioneritas dalam *mean* dapat diketahui melalui *Augmented Dickey Fuller Test* (ADF *Test*) yang ditampilkan dalam Tabel 4.19.

Tabel 4.19. Uji ADF Residual TSR *Outflow* Rp 100.000,-

Lag-order	d	Pvalue
5	0	0,010
5	1	0,010
5	12	0,013

Tabel 4.19 menunjukan bahwa residual stasioner dalam *mean* baik tanpa *differencing*, dengan *differencing* lag-1 maupun lag-12. Sehingga *differencing* tidak perlu dilakukan. Selanjutnya orde *p* dan *q* ARIMA diduga menggunakan ACF dan PACF Gambar 4.40

**Gambar 4.40.** ACF dan PACF Residual TSR *Outflow* Rp 100.000,-

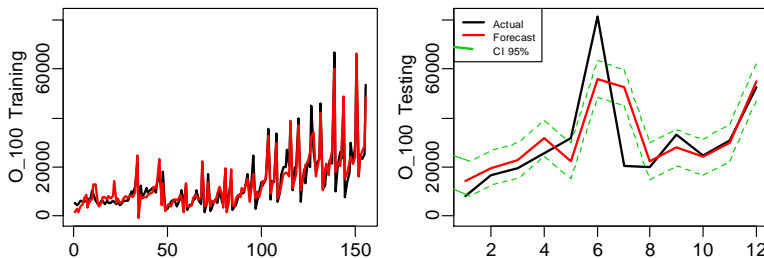
ACF menunjukan bahwa autokorelasi pada lag 5, 7, 10, 12 dan 15 signifikan. Sedangkan PACF menunjukan parsial autokorelasi yang signifikan adalah pada lag 7, 10, dan 12. Sehingga terdapat beberapa kemungkinan model yang akan dipilih

salah satu dengan kriteria residual *white noise*, paramater signifikan dan memiliki AIC terkecil. Kemungkinan model yang dapat terbentuk ditampilkan dalam Tabel 4.20.

Tabel 4.20. Kemungkinan Model ARIMA Residual TSR *Outflow* Rp 100.000,-

Model	White Noise	Normalitas	AIC
ARIMA([10,12],0,0)	Tidak	Ya	3034,82
ARIMA([10,12],0,[2])	Ya	Ya	3029,44
ARIMA(0,0,[12,24])	Tidak	Tidak	3048,80

Sehingga model yang dipilih sebagai model terbaik *outflow* Rp.100000,- adalah ARIMAX([10,12],0,[2]) dengan hasil *fits* dan *forecast* yang ditampilkan pada Gambar 4.41.



Gambar 4.41. Fits & Forecast ARIMAX Outflow Rp 100.000,-

Pemodelan menggunakan ARIMAX dilakukan dengan cara yang sama terhadap pecahan lain baik *inflow* maupun *outflow* sehingga diperoleh model ARIMAX terbaik untuk 7 pecahan *outflow* dan 7 pecahan *inflow* dalam Tabel 4.21.

Tabel 4.21. Model ARIMAX *Outflow* dan *Inflow*

Pecahan	Model	Residual	
		WN	Normal
Outflow 100	ARIMAX ([10,12],0,[2])	Ya	Normal
Outflow 50	ARIMAX ([12,13,23,26],0,0)	Ya	Normal
Outflow 20	TSR	Ya	Tidak
Outflow 10	TSR	Ya	Tidak
Outflow 5	ARIMAX ([1,12,23],0,0)	Ya	Tidak
Outflow 2	TSR	Ya	Tidak
Outflow 1	ARIMAX ([1,11,13],0,0)	Ya	Tidak

Pecahan	Model	Residual	
		WN	Normal
Inflow 100	ARIMAX (0,0,[1,12])	Ya	Tidak
Inflow 50	ARIMAX (0,0,[16,18,24,25])	Ya	Tidak
Inflow 20	ARIMAX ([1,2,6],0,[3,11,12,20])	Ya	Tidak
Inflow 10	ARIMAX (1,0,[1,11,12,23])	Ya	Tidak
Inflow 5	ARIMAX ([1,11,12,13],0,0)	Ya	Normal
Inflow 2	ARIMAX ([1,14],0,0)	Ya	Normal
Inflow 1	ARIMAX ([1,7,11],0,[4,12])	Ya	Normal

Berdasarkan ringkasan model terbaik ARIMAX yang telah diperoleh, dapat disimpulkan bahwa seluruh model ARIMAX untuk *inflow* dan *outflow* telah memenuhi asumsi *white noise* namun belum semua model memenuhi asumsi residual berdistribusi normal. Model yang diperoleh untuk masing-masing pecahan mengikuti persamaan

a. Persamaan ARIMAX *Outflow* Rp 100.000,-

$$\begin{aligned}
 Y_{1,t} = & 280,61t + 18941,9D_1 + 12308,4D_2 - 147,60tD_1 - \\
 & 147,35tD_2 - 17528,9M_{1,t} - 16331,7M_{2,t} - 18424,9M_{3,t} - \\
 & 15692,5M_{4,t} - 15242,3M_{5,t} - 12726,1M_{6,t} - 11210,6M_{7,t} - \\
 & 16193,9M_{8,t} - 12917,8M_{9,t} - 15518,6M_{10,t} - 23686M_{11,t} - \\
 & 8195,5M_{12,t} + 18198L_{1,t-1} + 3354,6L_{2,t-1} - 4043,8L_{3,t-1} + \\
 & 1272,7L_{4,t-1} + 2815,7L_{1,t} + 11505,2L_{2,t} + 16131,4L_{3,t} + \\
 & \quad (1 + 0,20B^2) \\
 & 17060,8L_{4,t} + \frac{(1 + 0,18B^{10} - 0,81B^{12})}{(1 + 0,18B^{10} - 0,81B^{12})} a_t
 \end{aligned}$$

b. Persamaan ARIMAX *Outflow* Rp 50.000,-

$$\begin{aligned}
 Y_{2,t} = & 116,19t + 13306,3D_1 + 8142,6D_2 - 78,96tD_1 - 109,26tD_2 + \\
 & -6254M_{1,t} - 4155,8M_{2,t} - 1575M_{3,t} - 1458,2M_{4,t} - \\
 & 848,37M_{5,t} + 1130,6M_{6,t} + 979,41M_{7,t} - 3020,7M_{8,t} - \\
 & 3064,4M_{9,t} - 2454,3M_{10,t} - 2864,5M_{11,t} + 6927,6M_{12,t} + \\
 & 20510,5L_{1,t-1} + 12848,2L_{2,t-1} + 14847L_{3,t-1} + 63,68291L_{4,t-1} \\
 & -2427,1L_{1,t} + 12848,2L_{2,t} + 14847L_{3,t} + 5867,9L_{4,t} + \\
 & (1 + 0,35B^{12} + 0,26B^{13} - 0,21B^{23} + 0,22B^{24})^{-1} a_t
 \end{aligned}$$

c. Persamaan ARIMAX *Outflow* Rp 20.000,-

$$Y_{3,t} = 2,25t + 1295,35D_1 + 561,85D_2 - 11,17tD_1 - 7,22tD_2 + \\ 299,21M_{1,t} + 127,36M_{2,t} + 200,43M_{3,t} + 293,38M_{4,t} + \\ 350,16M_{5,t} + 578,35M_{6,t} + 577,16M_{7,t} + 189,13M_{8,t} - \\ 84,35M_{9,t} + 23,04M_{10,t} + 131,02M_{11,t} + 556,74M_{12,t} + \\ 2039,46L_{1,t-1} + 1379,7L_{2,t-1} + 1463,27L_{3,t-1} + 459,48L_{4,t-1} - \\ 205,25L_{1,t} + 469,44L_{2,t} + 1192,6L_{3,t} + 1813,53L_{4,t}$$

d. Persamaan ARIMAX *Outflow* Rp 10.000,-

$$Y_{4,t} = 4,31t + 456,14D_1 + 164,17D_2 - 5,793tD_1 - 2,14tD_2 + \\ 78,27M_{1,t} - 18,08M_{2,t} + 9,37M_{3,t} + 65,71M_{4,t} + \\ 98M_{5,t} + 254,65M_{6,t} + 367,82M_{7,t} + 15,31M_{8,t} - \\ 286,03M_{9,t} - 399,98M_{10,t} - 249,25M_{11,t} + 100,5M_{12,t} + \\ 947,96L_{1,t-1} + 1473,46L_{2,t-1} + 1247,5L_{3,t-1} + 434,05L_{4,t-1} + \\ 204,53L_{1,t} + 400,88L_{2,t} + 960,4L_{3,t} + 1906,67L_{4,t}$$

e. Persamaan ARIMAX *Outflow* Rp 5.000,-

$$Y_{5,t} = 2,50t + 268,10D_1 + 182,23D_2 - 3,20tD_1 - 2,87tD_2 + \\ -88,07M_{1,t} - 80,48M_{2,t} - 44,51M_{3,t} + 42,49M_{4,t} + \\ 122,74M_{5,t} + 211,64M_{6,t} + 317,42M_{7,t} + 16,95M_{8,t} - \\ 214,24M_{9,t} - 319,57M_{10,t} - 241,26M_{11,t} - 65,77M_{12,t} + \\ 1620,7L_{1,t-1} + 1175,5L_{2,t-1} + 804,9L_{3,t-1} + 359,9L_{4,t-1} - \\ 104,9L_{1,t} + 164,5L_{2,t} + 521,2L_{3,t} + 903,9L_{4,t} + \\ (1 + 0,16B - 0,41B^{12} - 0,69B^{23})^{-1}a_t$$

f. Persamaan ARIMAX *Outflow* Rp 2.000,-

$$Y_{6,t} = 1,66t + 2,88D_2 - 0,79tD_2 + 72,07M_{1,t} + 66,29M_{2,t} + \\ 71,59M_{3,t} + 96,32M_{4,t} + 155,21M_{5,t} + 201,02M_{6,t} + \\ 76,26M_{7,t} - 182,38M_{8,t} + 2,94M_{9,t} + 13,88M_{10,t} + \\ 30,62M_{11,t} + 112,4M_{12,t} + 1480,55L_{2,t-1} + 772,72L_{3,t-1} + \\ 182,97L_{4,t-1} + 230,8L_{2,t} + 781,18L_{3,t} + 1540,713L_{4,t}$$

g. Persamaan ARIMAX *Outflow* Rp 1.000,-

$$Y_{7,t} = -0,795t + 37,04D_1 + 244,68D_2 + 1,23tD_1 - 2,85tD_2 + 107,10M_{1,t} + 93,02M_{2,t} + 94,72M_{3,t} + 94,93M_{4,t} + 95,58M_{5,t} + 97,04M_{6,t} + 84,53M_{7,t} + 127,49M_{8,t} + 139,53M_{9,t} + 147,94M_{10,t} + 108,07M_{11,t} + 112,92M_{12,t} + 276,08L_{1,t-1} + 104,67L_{2,t-1} + 24,83L_{3,t-1} + 49,63L_{4,t-1} - 65,62L_{1,t} + 113,62L_{2,t} + 136,95L_{3,t} + 248,49L_{4,t} (1 - 0,458B - 0,256B^{11} + 0,216B^{13})^{-1}a_t$$

h. Persamaan ARIMAX *Inflow* Rp 100.000,-

$$Y_{8,t} = 256,36t + 19230,5D_1 + 11120,4D_2 - 175,2tD_1 - 136,2tD_2 - 6456,7M_{1,t} - 14539,3M_{2,t} - 15531M_{3,t} - 15871,2M_{4,t} - 16037,5M_{5,t} - 16255,7M_{6,t} - 13395,6M_{7,t} - 11569,5M_{8,t} - 16290,5M_{9,t} - 15425,2M_{10,t} - 18088,4M_{11,t} - 20726,6M_{12,t} + 19767L_{1,t} + 11382,6L_{2,t} + 11380,7L_{3,t} - 3016,5L_{4,t} + 2824,2L_{1,t+1} + 2570,9L_{2,t+1} - 4453,2L_{3,t+1} + 16906,9L_{4,t+1} + (1 + 0,203B - 0,534B^{12})a_t$$

i. Persamaan ARIMAX *Inflow* Rp 50.000,-

$$Y_{9,t} = 109,6t + 11543,5D_1 + 4823,7D_2 - 24,2tD_1 - 70,2tD_2 + 5070,1M_{1,t} - 630,6M_{2,t} - 1130,6M_{3,t} - 1122,8M_{4,t} - 1960,2M_{5,t} - 1877,4M_{6,t} - 70,5M_{7,t} + 231,4M_{8,t} - 1746,2M_{9,t} - 464,2M_{10,t} - 3262,8M_{11,t} - 4342,5M_{12,t} + 14964,9L_{1,t} + 10575,3L_{2,t} + 5060,6L_{3,t} - 3479,1L_{4,t} + 2672,9L_{1,t+1} + 3329,5L_{2,t+1} + 5725,5L_{3,t+1} + 13086,5L_{4,t+1} + (1 + 0,221B^{16} + 0,195B^{18} + 0,379B^{24} + 0,205)a_t$$

j. Persamaan ARIMAX *Inflow* Rp 20.000,-

$$Y_{10,t} = 3,66t + 1310,4D_1 + 1158,4D_2 - 10,8tD_1 - 11,4tD_2 + 5070,1M_{1,t} - 630,6M_{2,t} - 1130,6M_{3,t} - 1122,8M_{4,t} - 1960,2M_{5,t} - 1877,4M_{6,t} - 70,5M_{7,t} + 231,4M_{8,t} - 1746,2M_{9,t} - 464,2M_{10,t} - 3262,8M_{11,t} - 4342,5M_{12,t} + 14964,9L_{1,t} + 10575,3L_{2,t} + 5060,6L_{3,t} - 3479,1L_{4,t} + 2672,9L_{1,t+1} + 3329,5L_{2,t+1} + 5725,5L_{3,t+1} + 13086,5L_{4,t+1} + \frac{(1 - 0,510B^3 - 0,294B^{11} - 0,380B^{12} + 0,304)B^{16}}{(1 - 0,219B - 0,250B^2 - 0,205B^6)}a_t$$

k. Persamaan ARIMAX *Inflow* Rp 10.000,-

$$Y_{11,t} = 3,601t + 394,96D_1 + 468,90D_2 - 7,65tD_1 - 5,83tD_2 + 333,29M_{1,t} + 208,46M_{2,t} + 189,44M_{3,t} + 129,70M_{4,t} - 60,65M_{5,t} - 16,11M_{6,t} + 27,49M_{7,t} + 157,51M_{8,t} + 208,33M_{9,t} + 219,22M_{10,t} + 207,22M_{11,t} + 180,38M_{12,t} + 250,04L_{1,t} + 37,89L_{2,t} - 90,40L_{3,t} - 148,38L_{4,t} + 139,24L_{1,t+1} + 172,04L_{2,t+1} + 395,26L_{3,t+1} + 358,91L_{4,t+1} (1 + 0,21B - 0,34B^{11} - 0,60B^{12} - 0,65B^{23})(1 - 0,574B)^{-1}a_t$$

l. Persamaan ARIMAX *Inflow* Rp 5.000,-

$$Y_{12,t} = 3,98t + 409,63D_1 + 188,32D_2 - 5,17tD_1 - 2,28tD_2 + 26,30M_{1,t} - 46,35M_{2,t} - 68,67M_{3,t} - 113,08M_{4,t} - 134,07M_{5,t} - 136,80M_{6,t} - 120,90M_{7,t} - 50,72M_{8,t} - 26,97M_{9,t} - 67,07M_{10,t} - 87,74M_{11,t} - 102,75M_{12,t} + 149,15L_{1,t} - 28,92L_{2,t} - 144,78L_{3,t} - 124,27L_{4,t} + 235,87L_{1,t+1} + 128,39L_{2,t+1} + 123,35L_{3,t+1} + 111,68L_{4,t+1} (1 + 0,371B + 0,510B^{11} + 0,406B^{12} - 0,334B^{13})^{-1}a_t$$

m. Persamaan ARIMAX *Inflow* Rp 2.000,-

$$Y_{13,t} = 1,54t - 207,69D_2 + 9,30tD_2 + 174,68M_{1,t} + 157,4M_{2,t} + 145,94M_{3,t} + 116,35M_{4,t} + 99,71M_{5,t} + 95,58M_{6,t} + 101,21M_{7,t} + 201,90M_{8,t} + 296,71M_{9,t} + 275,30M_{10,t} + 200,85M_{11,t} + 174,25M_{12,t} - 130,70L_{2,t} - 115,64L_{3,t} - 99,57L_{4,t} - 15,88L_{2,t+1} - 59,86L_{3,t+1} - 101,29L_{4,t+1} (1 + 0,444B - 0,442B^{14})^{-1}a_t$$

n. Persamaan ARIMAX *Inflow* Rp 1.000,-

$$Y_{14,t} = -0,929t + 9,86D_1 + 36,77D_2 + 1,04tD_1 - 0,22tD_2 + 204,32M_{1,t} + 181,33M_{2,t} + 176,09M_{3,t} + 153,76M_{4,t} + 142,53M_{5,t} + 130,86M_{6,t} + 134,73M_{7,t} + 134,99M_{8,t} + 132,16M_{9,t} + 148,07M_{10,t} + 170,13M_{11,t} + 182,98M_{12,t} - 7,80L_{1,t} - 21,50L_{2,t} - 37,17L_{3,t} - 57,52L_{4,t} + 41,07L_{1,t+1} + 34,39L_{2,t+1} + 6,57L_{3,t+1} - 8,82L_{4,t+1} \frac{(1 + 0,179B^4 - 0,496B^{12})}{(1 - 0,187B + 0,286B^7 - 0,309B^{11})}a_t$$

Kesalahan setiap model dapat diukur dengan RMSE dan sMAPE dari residual setiap model menggunakan persamaan (2.22) dan persamaan (2.23). Hasil RSME dan sMAPE yang diperoleh pada masing-masing model ditunjukkan pada Tabel 4.22.

Tabel 4.22. RMSE dan sMAPE ARIMAX Kajian Terapan

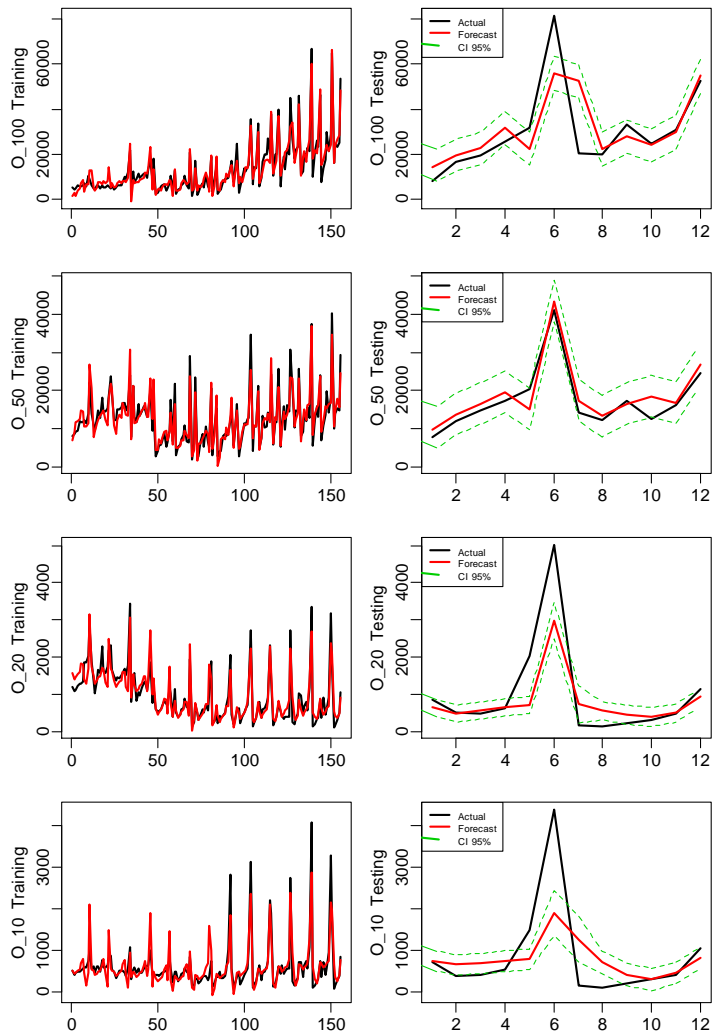
Model Pecahan	<i>Traning</i>		<i>Testing</i>	
	RMSE	sMAPE	RMSE	sMAPE
Outflow 100	3331,99	25,79	12602,52	25,18
Outflow 50	2519,50	17,35	2925,84	15,05
Outflow 20	268,58	23,03	736,43	46,58
Outflow 10	289,29	33,19	844,11	56,54
Outflow 5	179,99	37,14	289,34	50,19
Outflow 2	72,12	28,29	437,86	64,55
Outflow 1	30,55	56,27	72,38	175,53
Inflow 100	3887,80	28,07	5048,98	13,09
Inflow 50	1937,29	11,97	3028,40	15,40
Inflow 20	132,29	11,53	227,87	17,94
Inflow 10	90,92	11,88	170,10	11,68
Inflow 5	63,88	12,66	134,50	13,14
Inflow 2	27,52	21,75	64,40	20,19
Inflow 1	17,64	21,37	12,72	110,71

Hasil akurasi ARIMAX menunjukan bahwa secara umum, ARIMAX mampu meramalkan *outflow* dan *outflow* di Indonesia dengan cukup baik pada mayoritas pecahan kecuali *outflow* 200 dan 1000 serta *inflow* 1000. Perbandingan akurasi *testing* terhadap *training* dapat dilihat dalam Tabel 4.23.

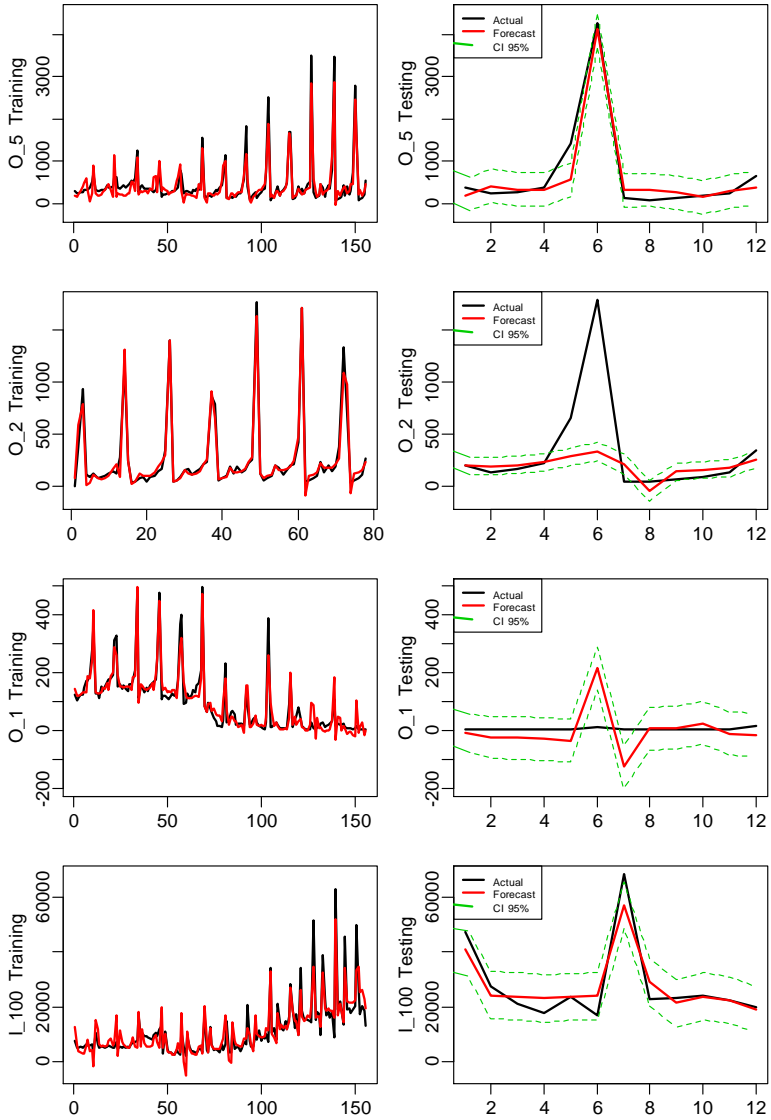
Tabel 4.23. Rasio RMSE dan sMAPE ARIMAX *Testing* Terhadap *Training*

Pecahan	Outflow		Inflow	
	RMSE	sMAPE	RMSE	sMAPE
100000	3,8	0,98	1,3	0,47
50000	1,2	0,87	1,6	1,29
20000	2,7	2,02	1,7	1,56
10000	2,9	1,70	1,9	0,98
5000	1,6	1,35	2,1	1,04
2000	6,1	2,28	2,3	0,93
1000	2,4	3,12	0,7	5,18

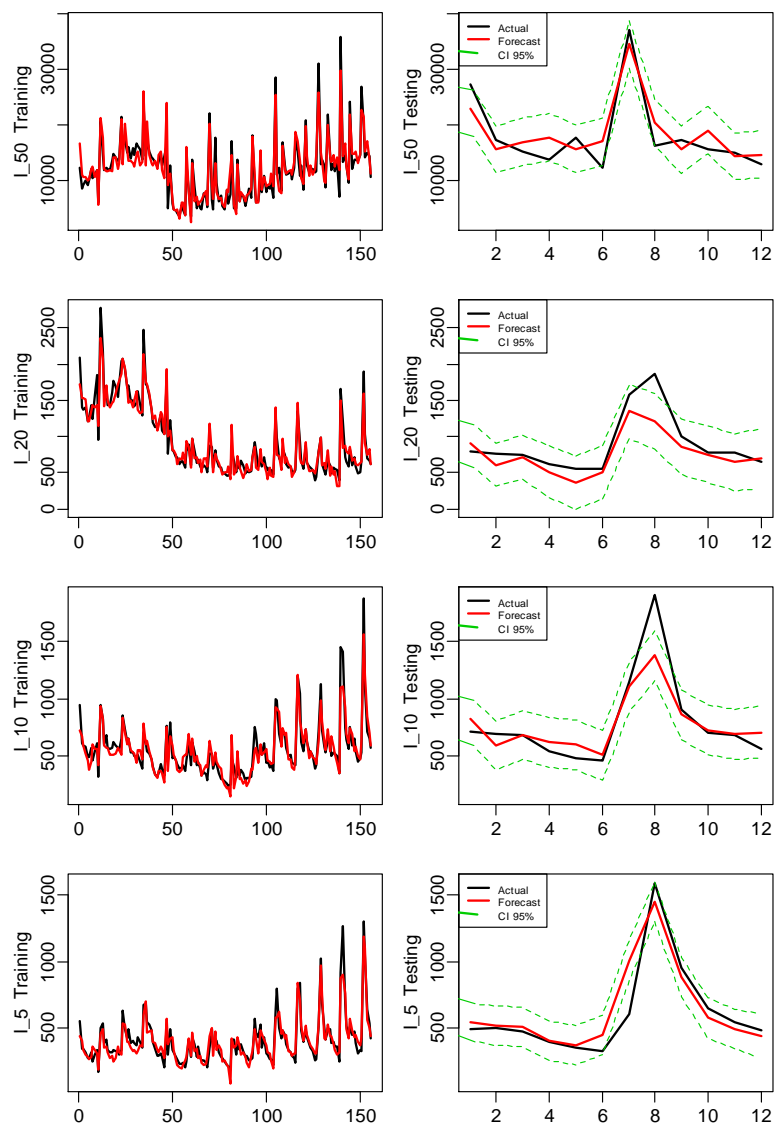
Perbandingan hasil ramalan model ARIMAX dengan data aktual (*testing*) untuk semua pecahan ditampilkan dalam Gambar 4.42.



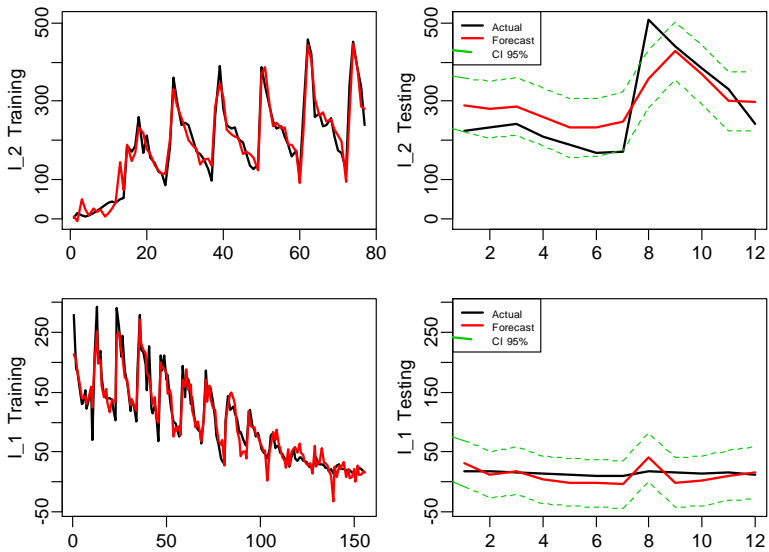
Gambar 4.42. Fits & Forecast ARIMAX Outflow dan Inflow



Gambar 4.42 (Lanjutan) *Fits & Forecast ARIMAX Outflow dan Inflow*



Gambar 4.42 (Lanjutan) Fits & Forecast ARIMAX Outflow dan Inflow

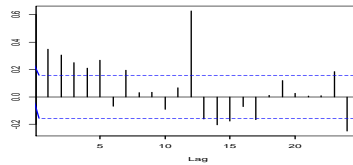


Gambar 4.42 (Lanjutan) *Fits & Forecast ARIMAX Outflow dan Inflow*

4.4 Peramalan *Inflow* dan *Outflow* Menggunakan DNN-1

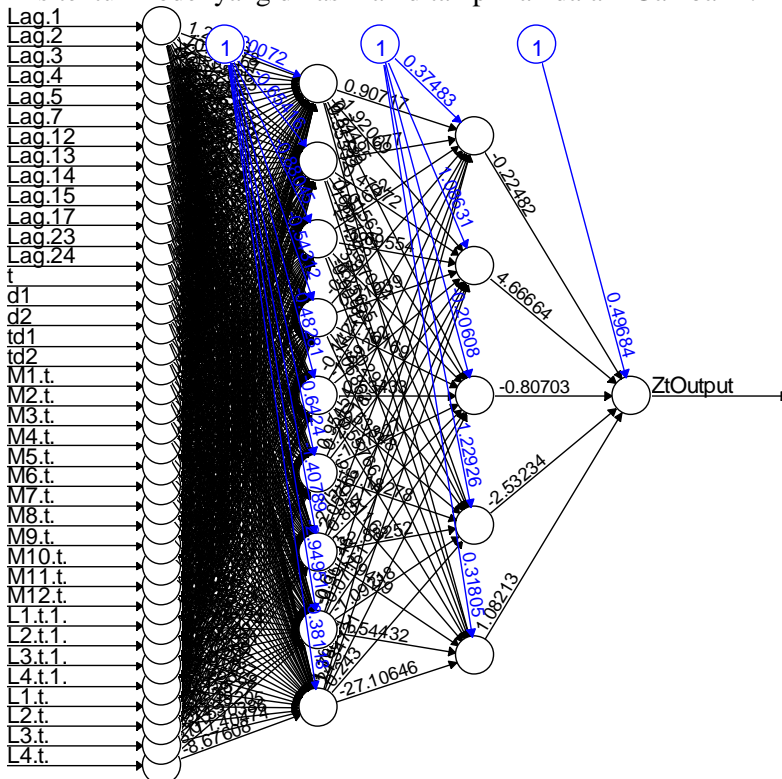
Peramalan menggunakan DNN-1 dilakukan melalui tahap *preprocessing* berupa standarisasi data, menentukan variabel input berdasarkan *lag* PACF yang signifikan, dan memilih kombinasi jumlah neuron optimum. Jumlah neuron yang dicobakan adalah 1 neuron sampai 10 neuron untuk setiap *hidden layer*. Setiap kombinasi jumlah neuron direplikasi sebanyak 10 kali. Replikasi dilakukan sebagai antisipasi adanya kemungkinan estimasi tidak konvergen untuk *set seed* tertentu, sehingga pada setiap kombinasi jumlah *neuron* akan dicobakan 10 *set seed* berbeda sebagai replikasi. Setiap 1 variabel dicobakan sebanyak 100 kombinasi model dikali 10 replikasi menjadi 1000 model, sehingga untuk 14 variabel menjadi 14000 model DNN-1.

Pada pecahan *outflow* Rp.100.000,- pemilihan input DNN *outflow* Rp.100.000,- menggunakan lag-PACF yang signifikan dalam Gambar 4.43 dan variabel *dummy* sesuai ARIMAX.



Gambar 4.43. PACF Outflow Rp 100.000,-

Gambar 4.43 menunjukan *lag* yang signifikan adalah *lag*-1 sampai 5, *lag* 12 sampai 15, *lag*-17, 23 dan 24. Kemudian dicobakan 100 kombinasi neuron berbeda, dan terpilih model terbaik dengan 9 *neuron* pada *hidden layer*-1 dan 5 *neuron* pada *hidden layer*-2. Arsitektur model yang dihasilkan ditampilkan dalam Gambar 4.44



Gambar 4.44. Arsitektur DNN *Outflow* Rp 100.000,-

Berdasarkan arsitektur dan bobot dari model tersebut, model dapat dituliskan dalam persamaan umum sebagai berikut

$$\hat{y}_{(k)} = f^0 \left[\sum_{l=1}^r \left[w_l^0 f_l^{h_2} \left(\sum_{j=1}^q w_j^{h_2} f_j^{h_1} \left(\sum_{i=1}^p w_{ji}^{h_1} x_{i(k)} + b_j^{h_1} \right) + b_l^{h_2} \right) + b^0 \right] \right]$$

sehingga jika disubstitusikan menggunakan bobot yang telah diestimasi, diperoleh model DNN untuk *outflow* Rp 100.000,- dalam persamaan

$$\hat{y}^* = 0,497 - 0,225 f_1^{h_2} + 4,667 f_2^{h_2} - 0,807 f_3^{h_2} - 2,532 f_4^{h_2} - 1,082 f_5^{h_2}$$

dengan bobot fungsi aktivasi *neuron* ke-1 sampai *neuron* ke-5 yang menuju *output layer* ditulis dalam persamaan

$$f_1^{h_2} = (1 + \exp(-(0,375 + 0,907 f_1^{h_1} - 0,168 f_2^{h_1} + \dots + 1,102 f_9^{h_1})))^{-1}$$

$$f_2^{h_2} = (1 + \exp(-(1,086 - 1,920 f_1^{h_1} + 0,475 f_2^{h_1} + \dots + 1,788 f_9^{h_1})))^{-1}$$

$$f_3^{h_2} = (1 + \exp(-(-0,206 + 0,775 f_1^{h_1} - 1,016 f_2^{h_1} + \dots + 24,347 f_9^{h_1})))^{-1}$$

$$f_4^{h_2} = (1 + \exp(-(1,229 + 0,818 f_1^{h_1} + 0,685 f_2^{h_1} + \dots + 0,245 f_9^{h_1})))^{-1}$$

$$f_5^{h_2} = (1 + \exp(-(0,318 + 0,353 f_1^{h_1} - 1,415 f_2^{h_1} + \dots - 27,106 f_9^{h_1})))^{-1}$$

serta bobot fungsi aktivasi *neuron* ke-1 sampai *neuron* ke-9 dari *hidden layer* 1 yang menuju *hidden layer* 2 ditulis dalam persamaan

$$f_1^{h_1} = (1 + \exp(-(0,201 + 1,221(x_1) - 0,365(x_2) + \dots - 2,129(x_{38}))))^{-1}$$

$$f_2^{h_1} = (1 + \exp(-(-0,654 + 1,259(x_1) + 0,317(x_2) + \dots - 6,978(x_{38}))))^{-1}$$

$$f_3^{h_1} = (1 + \exp(-(-0,880 - 0,237(x_1) + 1,599(x_2) + \dots - 3,108(x_{38}))))^{-1}$$

$$f_4^{h_1} = (1 + \exp(-(-0,543 + 1,681(x_1) - 1,441(x_2) + \dots + 4,999(x_{38}))))^{-1}$$

$$f_5^{h_1} = (1 + \exp(-(-0,483 - 1,259(x_1) - 2,095(x_2) + \dots - 4,339(x_{38}))))^{-1}$$

$$f_6^{h_1} = (1 + \exp(-(-0,642 - 1,010(x_1) - 0,252(x_2) + \dots + 4,368(x_{38}))))^{-1}$$

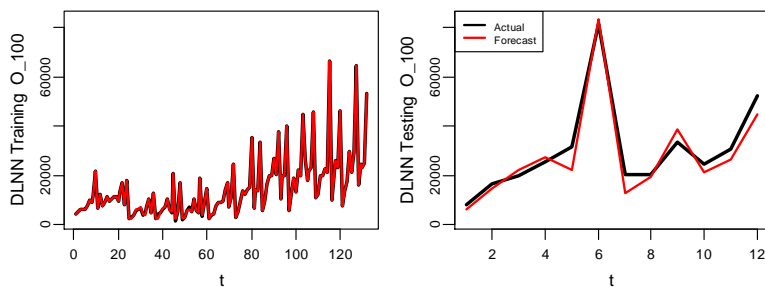
$$f_7^{h_1} = (1 + \exp(-(-1,408 + 0,341(x_1) + 0,397(x_2) + \dots + 3,198(x_{38}))))^{-1}$$

$$f_8^{h_1} = (1 + \exp(-(-2,949 - 1,535(x_1) + 1,513(x_2) + \dots - 3,016(x_{38}))))^{-1}$$

$$f_9^{h_1} = (1 + \exp(-(0,381 - 2,512(x_1) + 0,488(x_2) + \dots - 8,676(x_{38}))))^{-1}$$

Sedangkan variabel x menyatakan variabel 38 variabel input mulai dari input *lag*, variabel *dummy* hingga efek variasi kalender sesuai dengan input Gambar 4.45 Selanjutnya diperoleh nilai *fits*

dan *forecast* dari model yang telah diestimasi yang ditampilkan sebagai dalam Gambar 4.45



Gambar 4.45. Fits & Forecast DNN-1 Outflow Rp 100,000,- dengan cara yang sama, pemodelan DNN-1 dilakukan terhadap seluruh pecahan sehingga diperoleh daftar model terbaik DNN-1 yang ditampilkan dalam Tabel 4.24.

Tabel 4.24. Model DNN-1 *Outflow* dan *Inflow*

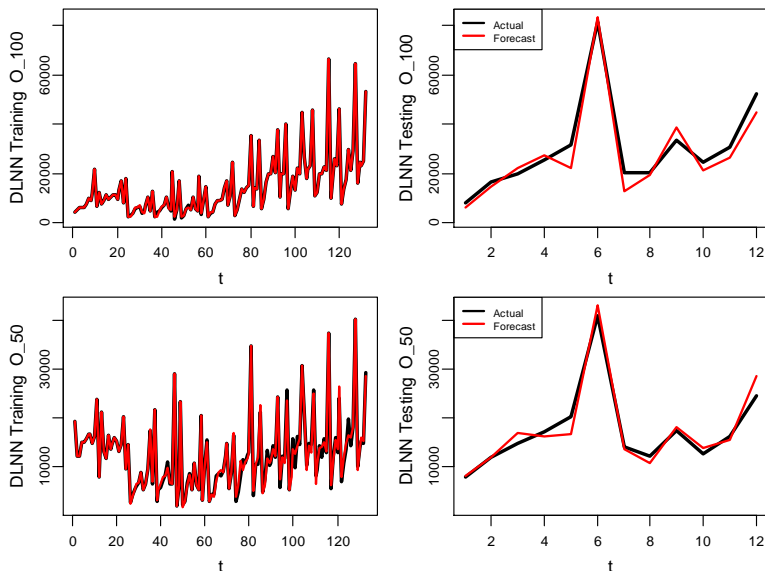
Pecahan	Neuron <i>Hidden</i> <i>Layer</i> Ke-		<i>Training</i>		<i>Testing</i>		
	1	2	RMSE	sMAPE	RMSE	sMAPE	
Outflow	100000	9	5	217,99	1,65	4750,18	16,89
	50000	7	2	915,78	5,34	1915,56	7,92
	20000	1	2	170,39	20,22	328,98	28,37
	10000	7	7	34,98	4,32	161,22	16,46
	5000	10	2	11,92	2,53	135,25	34,88
	2000	8	8	12,82	6,12	73,44	31,31
	1000	10	8	1,45	6,32	2,96	43,69
Inflow	100000	7	10	333,22	2,35	3933,04	11,78
	50000	7	3	264,97	0,97	1867,02	10,12
	20000	6	9	25,27	1,97	58,30	6,48
	10000	2	5	62,57	8,68	45,44	5,53
	5000	5	5	16,06	2,72	56,24	4,64
	2000	10	10	1,86	0,32	15,69	4,45
	1000	10	7	1,35	2,16	1,67	10,21

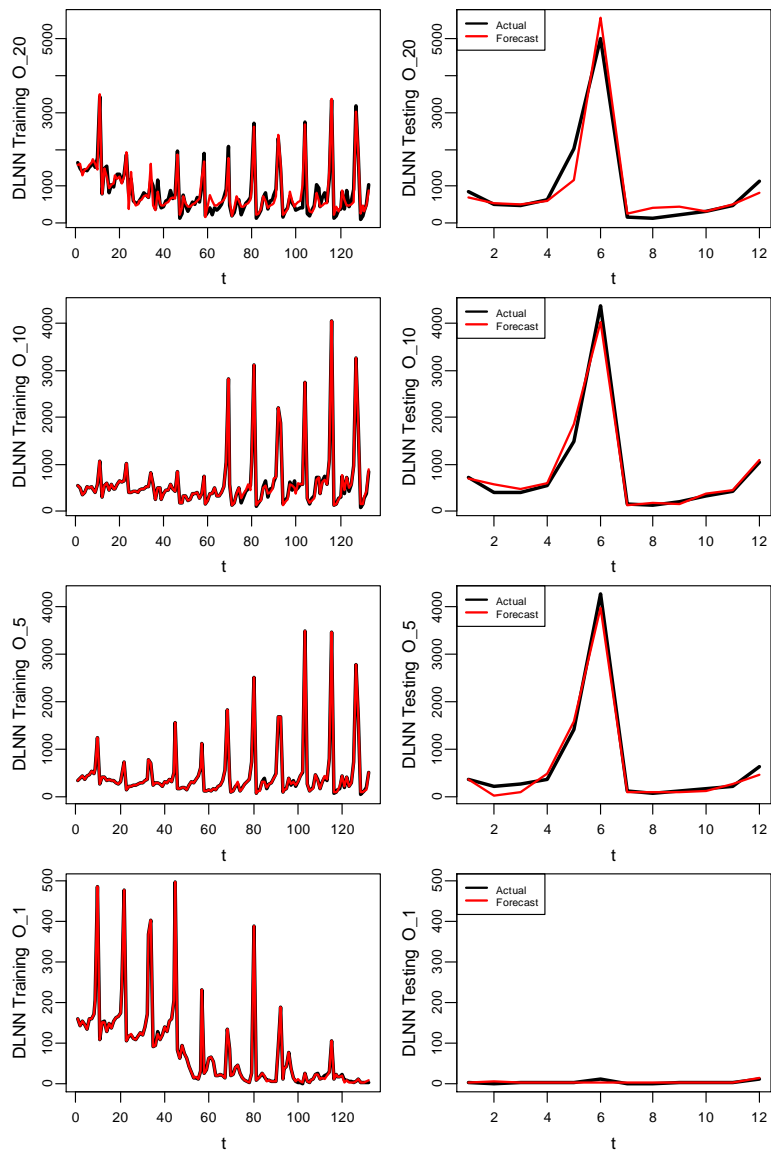
Perbandingan akurasi *testing* terhadap *training* DNN ditampilkan dalam Tabel 4.25.

Tabel 4.25. Rasio RMSE dan sMAPE DNN-1 *Testing* Terhadap *Training*

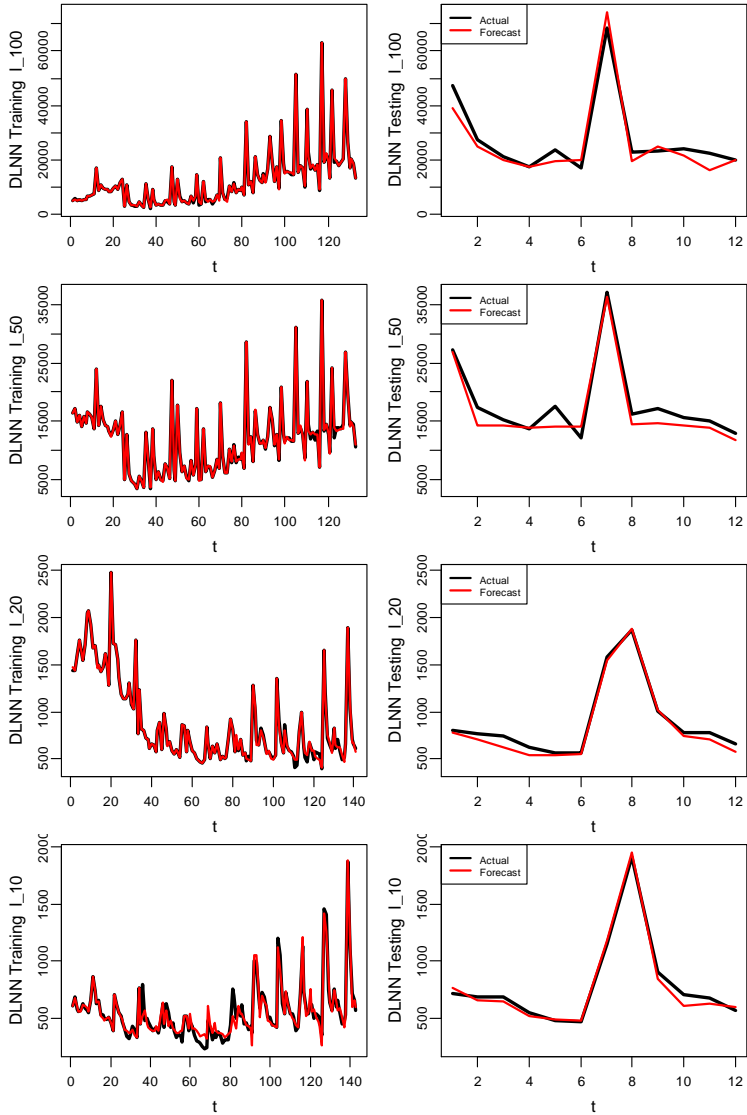
Pecahan	Outflow		Inflow	
	RMSE	sMAPE	RMSE	sMAPE
100000	21,8	10,25	11,8	5,02
50000	2,1	1,48	7,0	10,44
20000	1,9	1,40	2,3	3,29
10000	4,6	3,81	0,7	0,64
5000	11,3	13,80	3,5	1,71
2000	5,7	5,12	8,4	13,91
1000	2,0	6,92	1,2	4,73

Tabel 4.25 menunjukkan bahwa perubahan tingkat kesalahan dari *training* ke *testing* cenderung membesar (namun masih wajar), karena rasio meunjukkan nilai lebih dari 1. Selanjutnya, dengan arsitektur DNN-1 terbaik masing-masing pecahan, diperoleh hasil *fits* dan *forecast* model DNN-1 terbaik yang ditampilkan dalam Gambar 4.46.

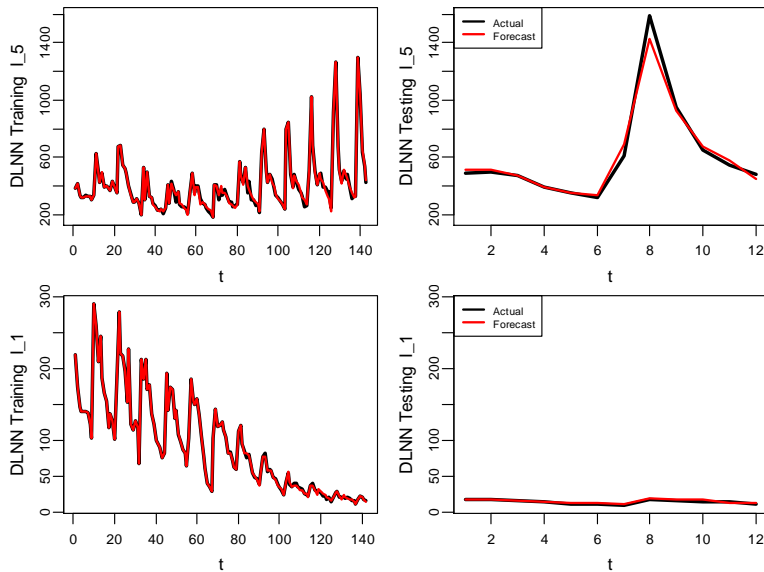
**Gambar 4.46.** Fits & Forecast DNN-1 Outflow dan Inflow



Gambar 4.46 (Lanjutan) *Fits & Forecast* DNN-1 Outflow dan Inflow



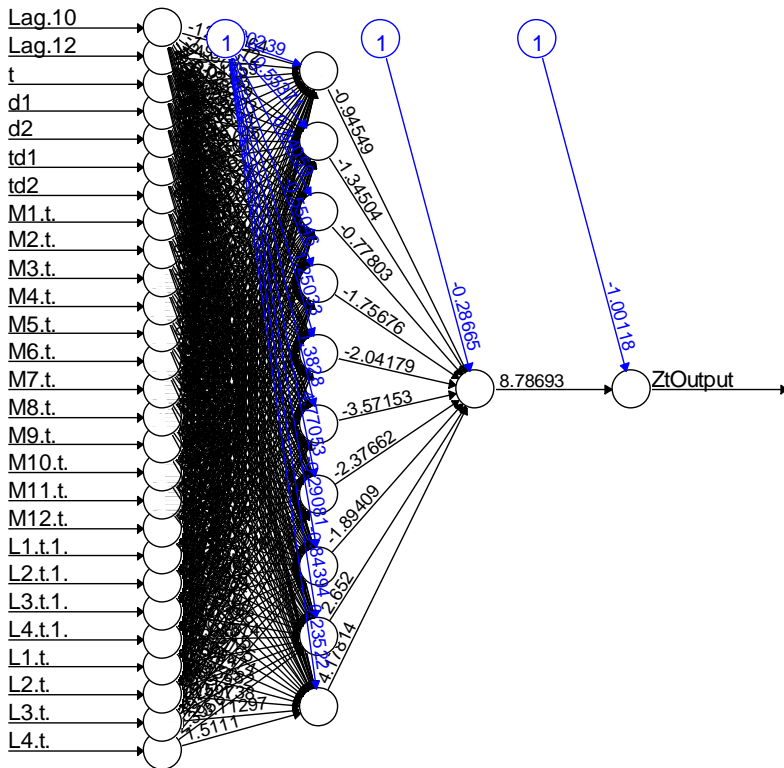
Gambar 4.46 (Lanjutan) *Fits & Forecast DNN-1 Outflow dan Inflow*



Gambar 4.46 (Lanjutan) *Fits & Forecast* DNN-1 *Outflow* dan *Inflow*

4.5 Peramalan *Inflow* dan *Outflow* Menggunakan DNN-2

Peramalan DNN-2 menggunakan tahapan yang sama dengan DNN-1 namun input lag yang digunakan adalah lag yang bersesuaian dengan orde AR model ARIMAX. Pada pecahan *outflow* Rp.100.000,- misalnya, input yang digunakan adalah variabel *dummy* serta variabel lag-10 dan lag-12 karena model ARIMAX yang terbentuk adalah $ARIMAX([10,12],0,[2])$. Sehingga dengan mencoba 1 sampai 10 neuron pada setiap *hidden layer*, diperoleh arsitektur DNN-2 terbaik untuk *outflow* Rp.100.000,- seperti pada Gambar 4.47. Penulisan model DNN-2 dari bobot yang telah diestimasi juga menggunakan cara yang sama seperti pada model DNN-1.

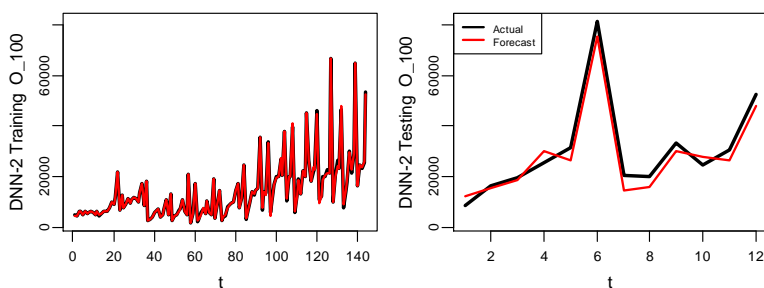


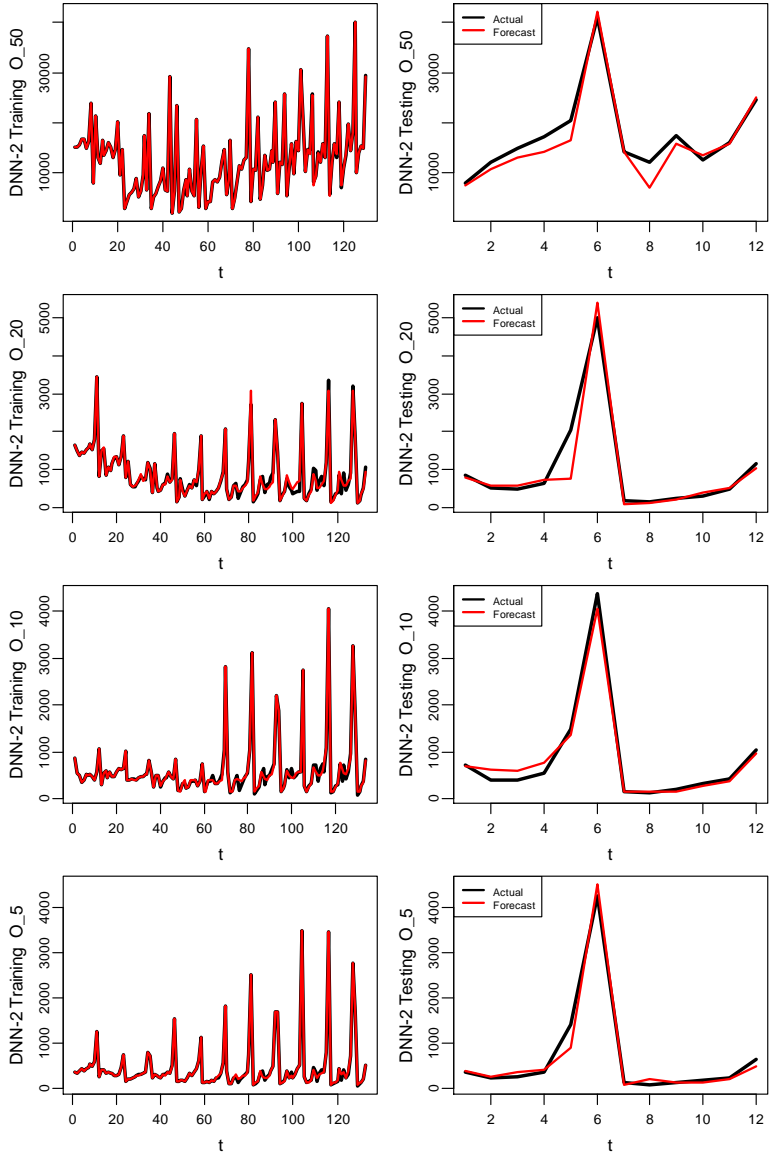
Gambar 4.47. Arsitektur DNN-2 Terbaik *Outflow* Rp.100.000,-
 Prosedur yang sama dilakukan pada semua pecahan untuk
 mendapatkan model DNN-2 terbaik dari setiap pecahan. Model
 DNN-2 untuk setiap pecahan ditampilkan dalam tabel berikut

Tabel 4.26. Model DNN-2 Terbaik *Outflow* dan *Inflow*

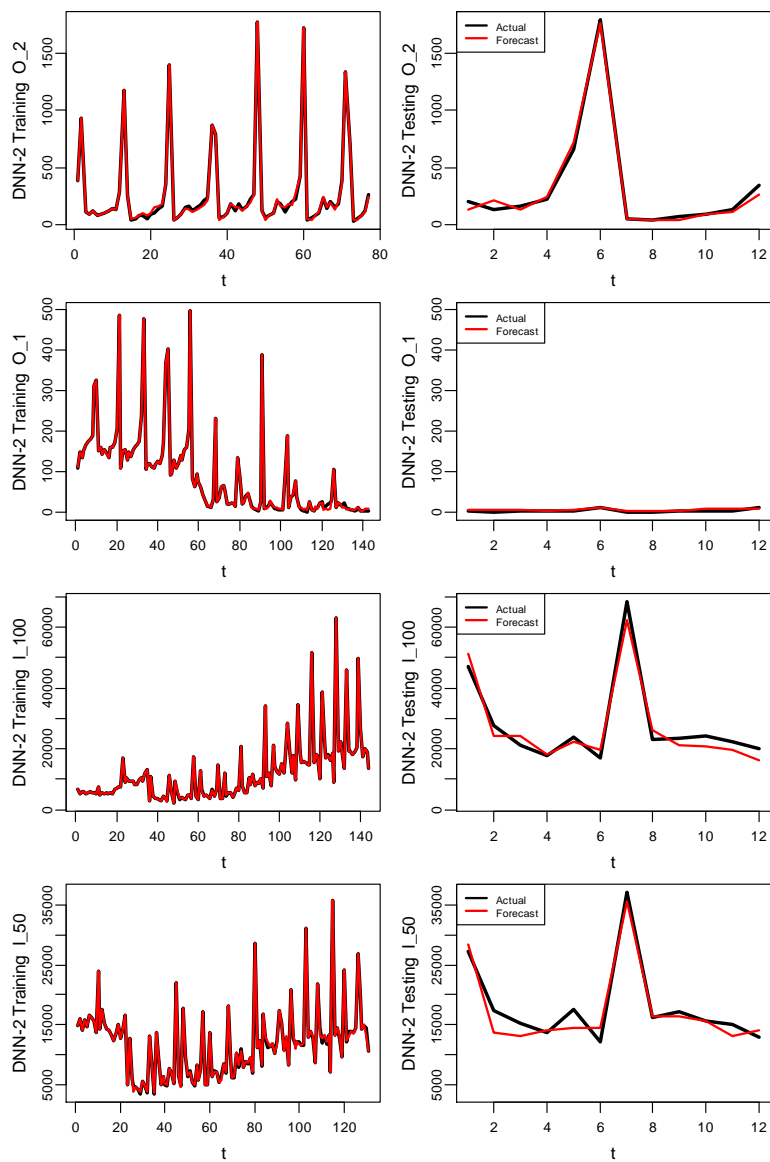
Pecahan	Neuron <i>Hidden Layer</i> Ke-		RMSE	
	1	2	<i>Training</i>	<i>Testing</i>
Outflow	100rb	10	478.11	4217.85
	50rb	10	130.68	2242.63
	20rb	5	95.23	391.23
	10rb	5	58.58	152.18
	5rb	8	25.76	178.95
	2rb	10	16.24	46.05
	1rb	8	2.51	3.69
Inflow	100rb	10	92.79	3366.14
	50rb	9	274.35	1845.90
	20rb	7	24.24	78.53
	10rb	8	22.40	96.47
	5rb	8	3.76	60.65
	2rb	8	0.72	17.21
	1rb	8	1.01	1.71

Tabel 4.26 menunjukan bahwa mayoritas jumlah neuron pada *hidden layer* pertama cenderung banyak (lebih dari 5) sedangkan jumlah neuron pada *hidden layer* ke-2 relatif bervariasi mulai dari 1 sampai 10. Berdasarkan model terbaik yang diperoleh, hasil *fits* dan *forecast* ditampilkan dalam gambar berikut bersama dengan data aktual untuk mengetahui kemampuan model dalam menangkap pola data.

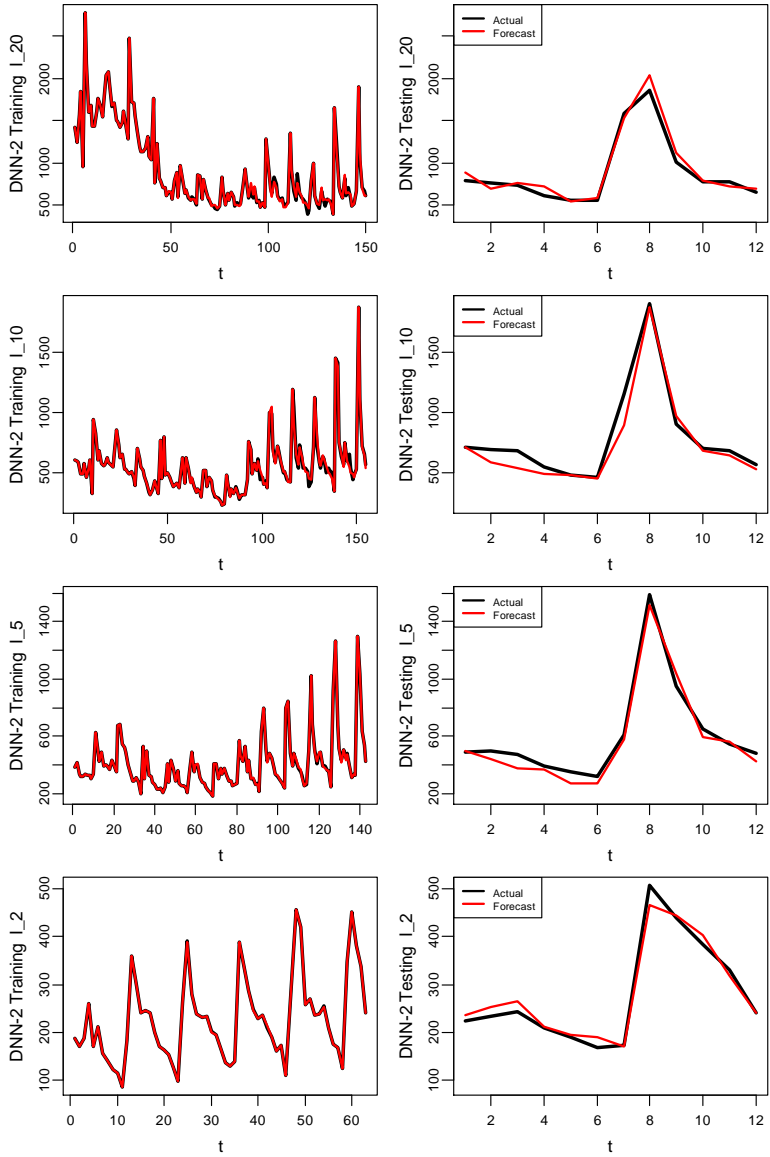
**Gambar 4.48** (Lanjutan) Fits & Forecast DNN-2 Outflow dan Inflow



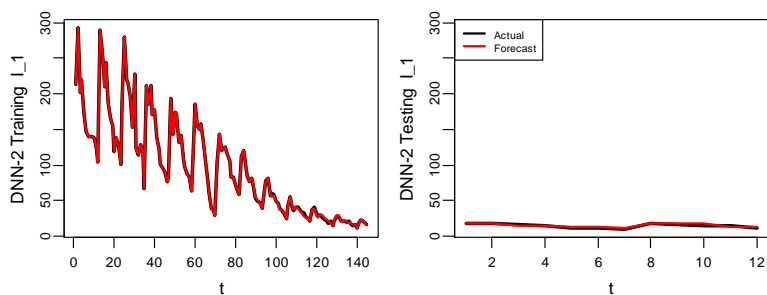
Gambar 4.48 (Lanjutan) Fits & Forecast DNN-2 Outflow dan Inflow



Gambar 4.48 (Lanjutan) Fits & Forecast DNN-2 Outflow dan Inflow



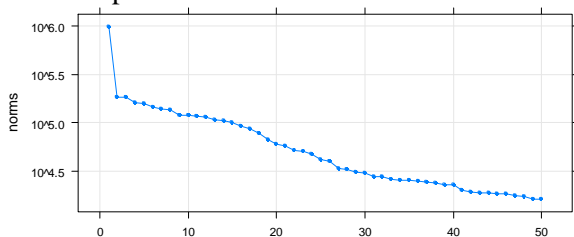
Gambar 4.48 (Lanjutan) Fits & Forecast DNN-2 Outflow dan Inflow



Gambar 4.48. (Lanjutan) Fits & Forecast DNN-2 Outflow dan Inflow

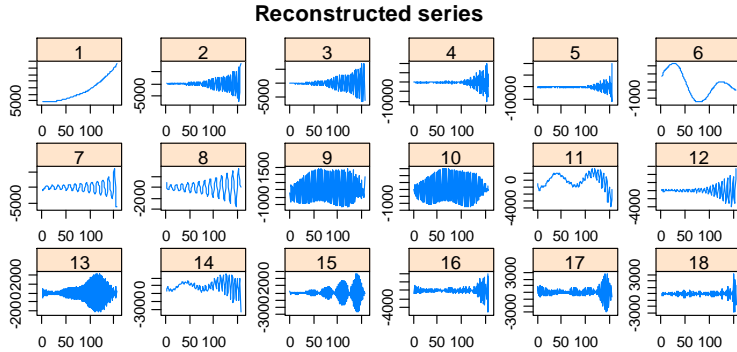
4.6 Peramalan *Inflow* dan *Outflow* Menggunakan SSA-DNN

Pemodelan menggunakan SSA-DNN dilakukan secara agregat pada setiap komponen yang terekonstruksi. Rekonstruksi dilakukan dengan mengelompokkan *eigen triples* yang berpola *trend* dan musiman menjadi 2 komponen. Selanjutnya komponen *trend* dimodelkan menggunakan regresi polinomial hingga orde 3 dengan variabel prediktor t . Sedangkan komponen musiman dimodelkan menggunakan regresi *dummy*. Pada *outflow* pecahan Rp.100.000,-, pengelompokan *eigen triples* dilakukan berdasarkan plot *eigenvalues* pada Gambar 4.49



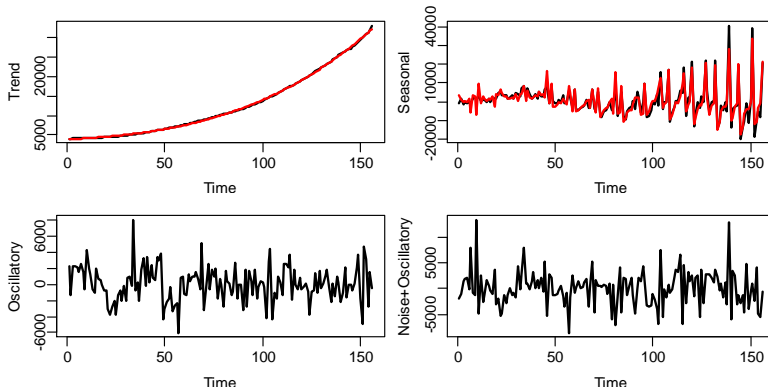
Gambar 4.49. Plot Eigenvalues Outflow Rp.100000,-

Komponen *noise* diidentifikasi melalui perubahan nilai *eigenvalue* yang turun lambat. Gambar 4.49 menunjukkan penurunan lambat terjadi mulai *eigenvalue* ke-19 dan seterusnya. Sehingga *eigen triples* ke-1 hingga ke-18 dikelompokkan kedalam *trend* dan musiman. Identifikasi dilakukan dengan rekonstruksi *eigen triples* secara individu seperti pada Gambar 4.50



Gambar 4.50. Rekonstruksi Individu *Outflow* Rp 100.000,-

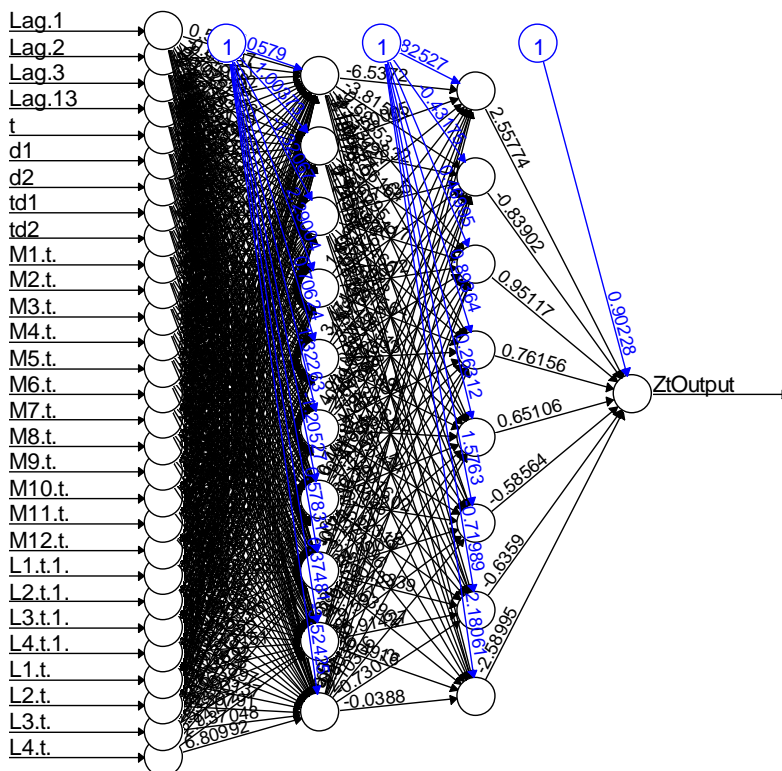
Gambar 4.50 menunjukkan bahwa komponen pertama memiliki pola *trend* sehingga akan dimodelkan dengan regresi polinomial, sedangkan komponen ke-2 hingga ke-18 memiliki pola musiman sehingga akan direkonstruksi ulang menjadi 1 komponen dan dimodelkan menggunakan regresi *dummy*. Hasil rekonstruksi ulang dan pemodelan ditampilkan dalam Gambar 4.51



Gambar 4.51. Rekonstruksi & Pemodelan Agregat Rp.10000,-

Gambar 4.51 menunjukkan bahwa pada *outflow* Rp 100.000,- pola *trend* dan musiman dapat dimodelkan dengan regresi dengan cukup baik sehingga nilai *fits* regresi (plot berwarna merah) dapat mengikuti plot komponen *trend* dan musiman hasil rekonstruksi SSA. Komponen *noise* ditambahkan dengan residual dari regresi

untuk dimodelkan dengan DNN dengan prosedur seperti DNN-1 sehingga diperoleh arsitektur model terbaik dalam Gambar 4.52.



Gambar 4.52. Arsitektur DNN dari SSA-DNN Outflow Rp 100.000,-

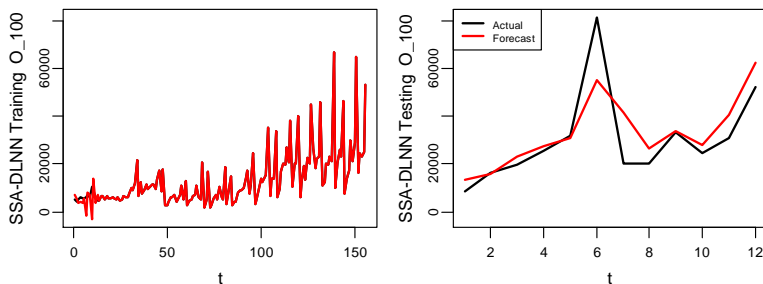
Gambar 4.52 menunjukkan model DNN terbaik untuk *noise* SSA. Input variabel dummy tren, musiman dan variasi kalender tetap diikutsertakan dalam DNN untuk mengantisipasi adanya pola tren dan musiman yang tidak terekonstruksi dengan baik oleh SSA atau tidak tertangkap oleh pemodelan regresi polinomial dan regresi *dummy*. Model tersebut memiliki 10 *neuron* pada *hidden layer-1* dan 8 *neuron* pada *hidden layer-2*. Hasil *fits* dan *forecast* DNN kemudian ditambahkan dengan hasil *fits* dan *forecast* dari regresi polinomial dan regresi *dummy* sehingga diperoleh *fits* dan

forecast SSA-DNN. Prosedur SSA-DNN yang sama diterapkan kepada semua pecahan *inflow* dan *outflow*. Model DNN dari *noise* SSA dari semua pecahan ditampilkan dalam Tabel 4.27.

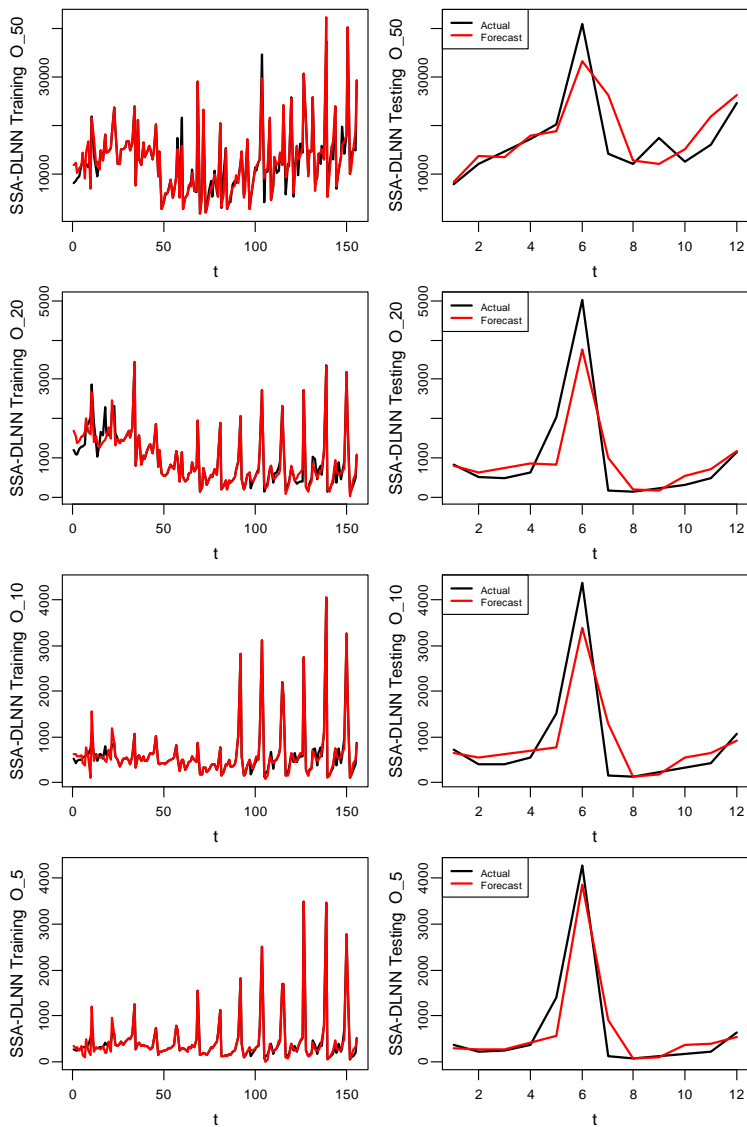
Tabel 4.27. Model Terbaik SSA-DNN *Inflow* dan *Outflow*

Pecahan	H-1	H-2	Training		Testing		
			RMSE	sMAPE	RMSEP	sMAPEP	
Outflow	100	10	8	1383,20	4,72	10964,71	22,59
	50	9	8	1653,04	7,91	4874,09	18,34
	20	8	1	155,52	10,91	576,25	41,83
	10	10	6	105,46	10,57	502,54	40,54
	5	9	10	81,09	13,36	365,52	38,24
	2	6	2	36,14	13,27	342,99	70,10
	1	10	3	9,26	5,02	56,56	162,13
Inflow	100	7	4	1735,79	9,29	5048,41	13,86
	50	2	8	1618,89	9,20	1795,54	9,58
	20	1	10	157,94	11,44	83,73	9,42
	10	7	4	53,74	3,61	160,11	14,07
	5	9	10	36,26	5,10	76,15	9,67
	2	10	8	14,80	17,34	36,16	10,40
	1	9	3	8,68	1,54	8,60	50,34

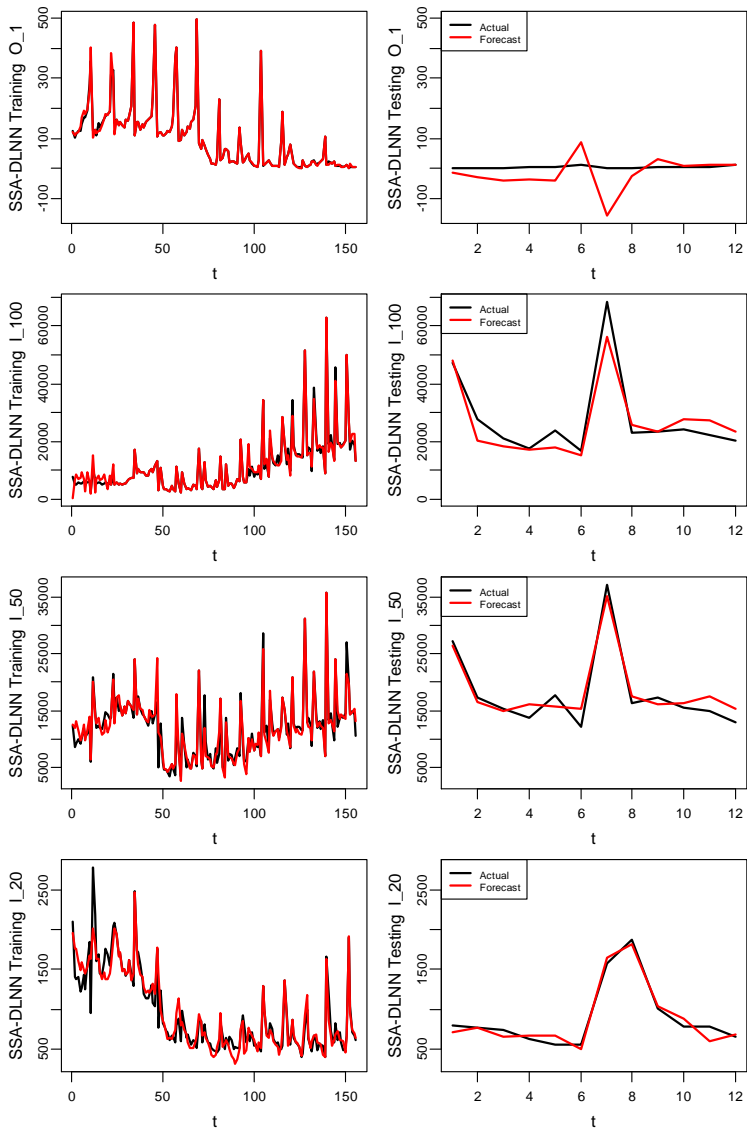
Berdasarkan model terbaik yang didapatkan, dapat diperoleh *nilai fits* dan *forecast* SSA-DNN untuk 12 periode pada masing-masing pecahan yang ditampilkan dalam Gambar 4.53.



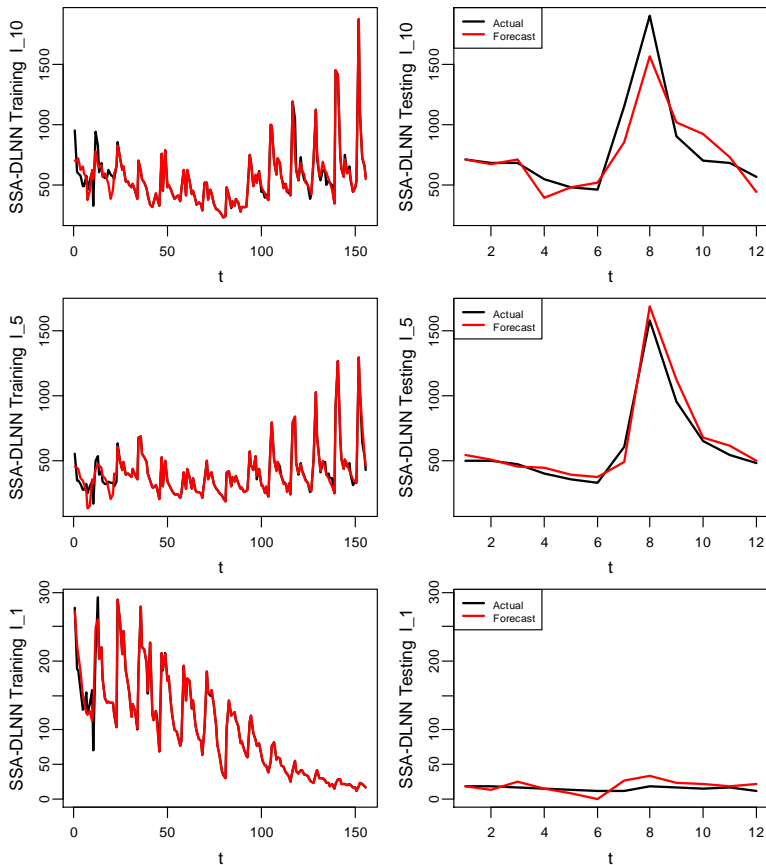
Gambar 4.53. Fits & Forecast SSA-DNN Outflow dan Inflow



Gambar 4.53 (Lanjutan) *Fits & Forecast SSA-DNN Outflow dan Inflow*



Gambar 4.53 (Lanjutan) *Fits & Forecast SSA-DNN Outflow dan Inflow*



Gambar 4.53 (Lanjutan) *Fits & Forecast SSA-DNN Outflow dan Inflow*

Secara umum, hasil *forecast* data *testing* SSA-DNN dapat mengikuti pola tren, musiman dan variasi kalender yang terdapat pada *outflow* dan *inflow* semua pecahan. Puncak ramalan dan pola naik turun ramalan juga sesuai dengan data *testing* pada hampir semua pecahan. Secara visual, akurasi SSA-DNN dapat dikatakan bagus.

4.7 Perbandingan Model ARIMAX, DNN dan SSA-DNN

Perbandingan kriteria RMSEP dari model ARIMAX, DNN-1, DNN-2 dan SSA-DNN dalam memprediksi *outflow* ditampilkan dalam Tabel 4.28.

Tabel 4.28. RMSEP *Outflow* Model ARIMAX, DNN-1, DNN-2 & SSA-DNN

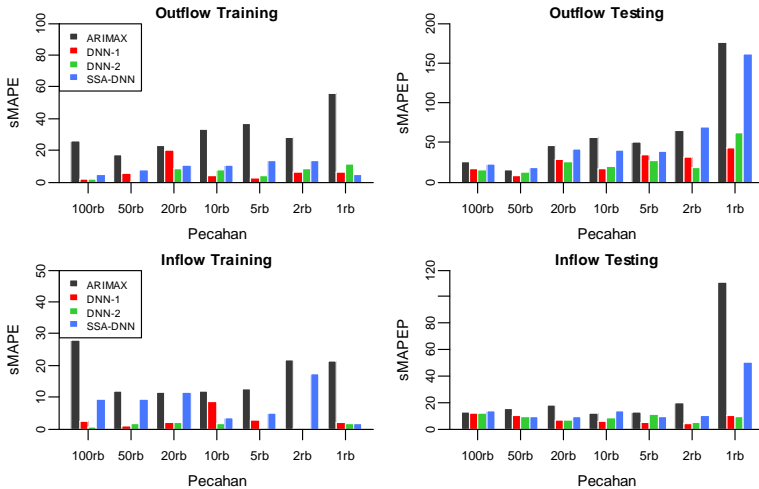
Pecahan	RMSEP			
	ARIMAX	DNN-1	DNN-2	SSA-DNN
100rb	12602.5	4750.2	4217.8	10964.7
50rb	2925.8	1915.6	2242.6	4874.1
20rb	736.4	329.0	391.2	576.2
10rb	844.1	161.2	152.2	502.5
5rb	289.3	135.3	178.9	365.5
2rb	437.9	73.4	46.0	343.0
1rb	72.4	3.0	3.7	56.6

Sedangkan perbandingan RMSEP untuk data *inflow* semua pecahan ditampilkan dalam Tabel 4.29.

Tabel 4.29. RMSEP *Inflow* Model ARIMAX, DNN-1, DNN-2 & SSA-DNN

Pecahan	RMSEP			
	ARIMAX	DNN-1	DNN-2	SSA-DNN
100rb	5049.0	3933.0	3366.1	5048.4
50rb	3028.4	1867.0	1845.9	1795.5
20rb	227.9	58.3	78.5	83.7
10rb	170.1	45.4	96.5	160.1
5rb	134.5	56.2	60.7	76.2
2rb	64.4	15.7	17.2	36.2
1rb	12.7	1.7	1.7	8.6

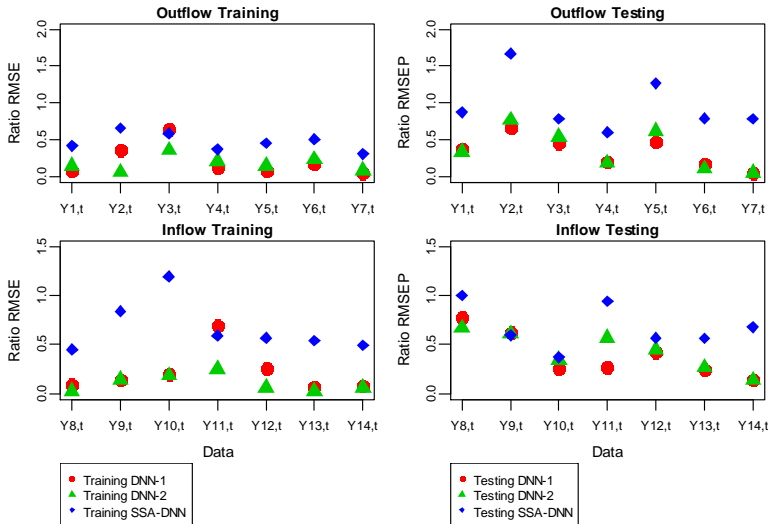
Kriteria lain yang digunakan sebagai bahan evaluasi kebaikan model selain RMSE adalah sMAPE. sMAPE lebih mudah untuk diinterpretasikan karena range nilai yang mungkin adalah antara 0 sampai 200, sedangkan RMSE memiliki range antara 0 sampai tak hingga sehingga relatif lebih sulit untuk diinterpretasikan. Namun, RMSEP tetap digunakan sebagai kriteria utama dalam pemilihan model terbaik. Perbandingan sMAPE dan sMAPEP dari ARIMAX, DNN-1, DNN-2 dan SSA_DNN ditampilkan dalam Gambar 4.54.



Gambar 4.54. sMAPE & sMAPEP Model ARIMAX, DNN-1, DNN-2 & SSA-DNN

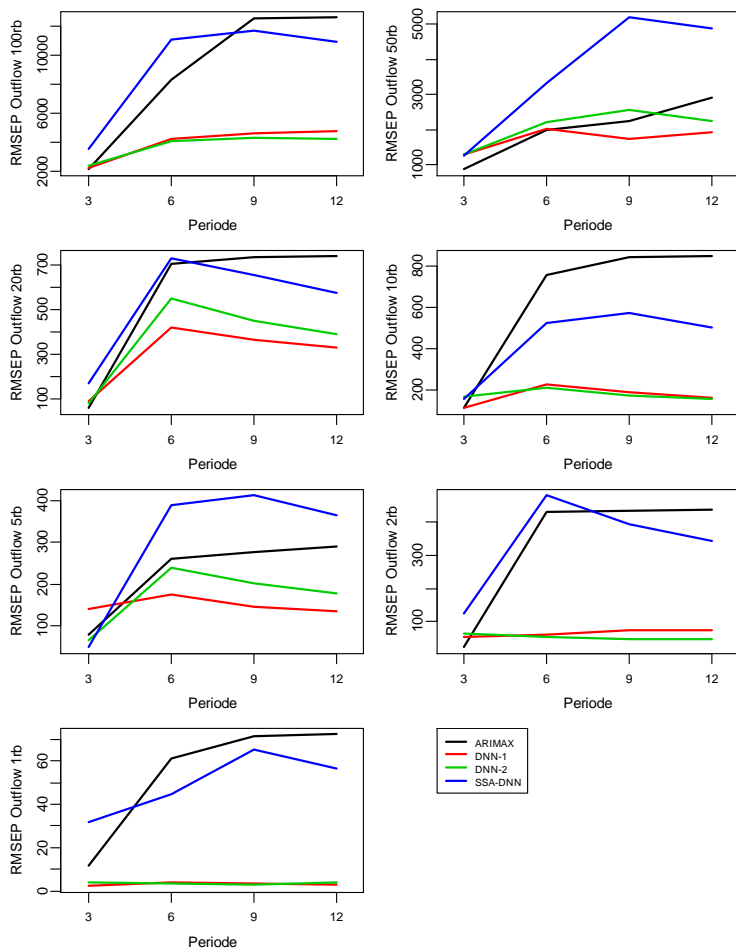
Kriteria sMAPE menunjukan hal yang tidak jauh berbeda dengan RMSE bahwa performa ARIMAX dalam memodelkan data *outflow* dan *inflow* tidak sebaik performa DNN-1, DNN-2 maupun SSA-DNN. Model yang memiliki akurasi terbaik adalah model DNN-1 dan DNN-2 baik pada *outflow* maupun *inflow* namun dominasi model DNN-1 terutama pada data *testing* masih lebih dominan. Sementara itu, model *hybrid* SSA-DNN justru tidak menunjukan akurasi yang lebih baik secara signifikan dibandingkan dengan model yang lebih sederhana. SSA-DNN hanya unggul pada *inflow* Rp 50.000,- baik dilihat dari RMSEP maupun sMAPEP. Lago (2018) juga membandingkan antara model *hybrid* dengan model yang lebih sederhana dan menyimpulkan bahwa model *hybrid* tidak lebih baik dalam memodelkan harga listrik yang memiliki pola musiman. Selain itu, penelitian tersebut juga menyimpulkan bahwa model *machine learning* memiliki akurasi lebih baik dibandingkan model statistika seperti ARIMAX.

Rasio kriteria RMSE dan RMSEP dari model DNN-1, DNN-2 dan SSA-DNN terhadap ARIMAX ditampilkan dalam Gambar 4.55 untuk mempermudah melihat perbandingan antar metode.



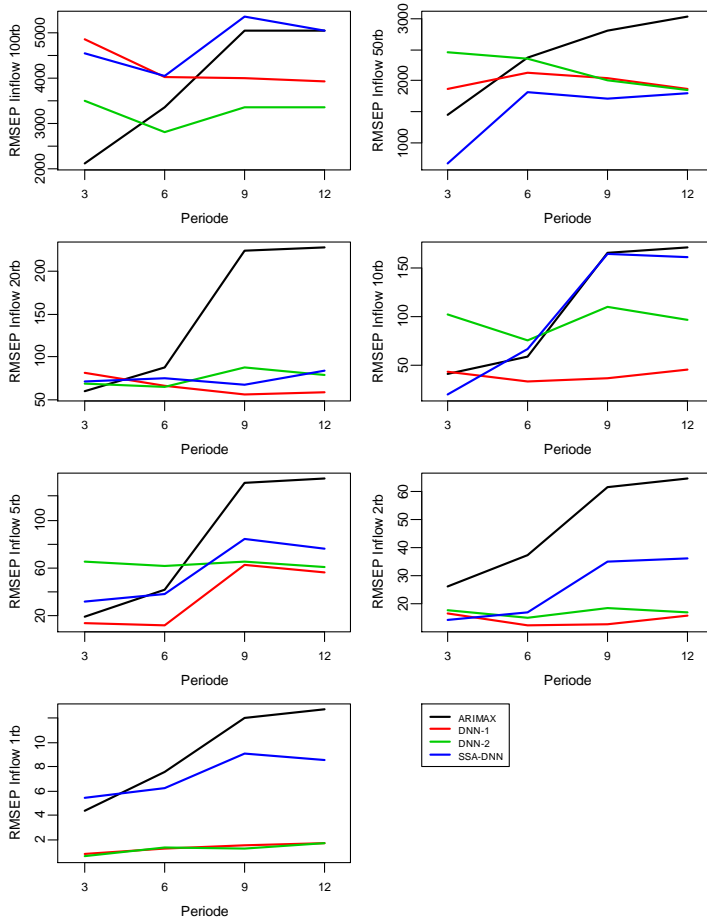
Gambar 4.55. Rasio RMSE Model DNN-1, DNN-2 & SSA-DNN Terhadap ARIMAX

Kebaikan model tertentu dalam melakukan peramalan juga akan berbeda-beda tergantung pada panjang periode ramalan. Umumnya, metode *machine learning* memiliki akurasi yang lebih besar untuk ramalan jangka pendek, sedangkan metode klasik seperti ARIMAX akan memiliki akurasi lebih baik walaupun periode ramalan lebih panjang jika pola yang terjadi pada data relatif konstan. Oleh karena itu, perbandingan akurasi antar metode pada ramalan 3 periode, 6 periode, 9 periode dan 12 periode ditampilkan dalam Gambar 4.56 untuk melihat seberapa jauh model dapat melakukan *forecasting* dengan baik.



Gambar 4.56. Perbandingan RMSEP ARIMAX, DNN-1, DNN-2 dan SSA-DNN Data *Outflow* Menurut Periode Ramalan

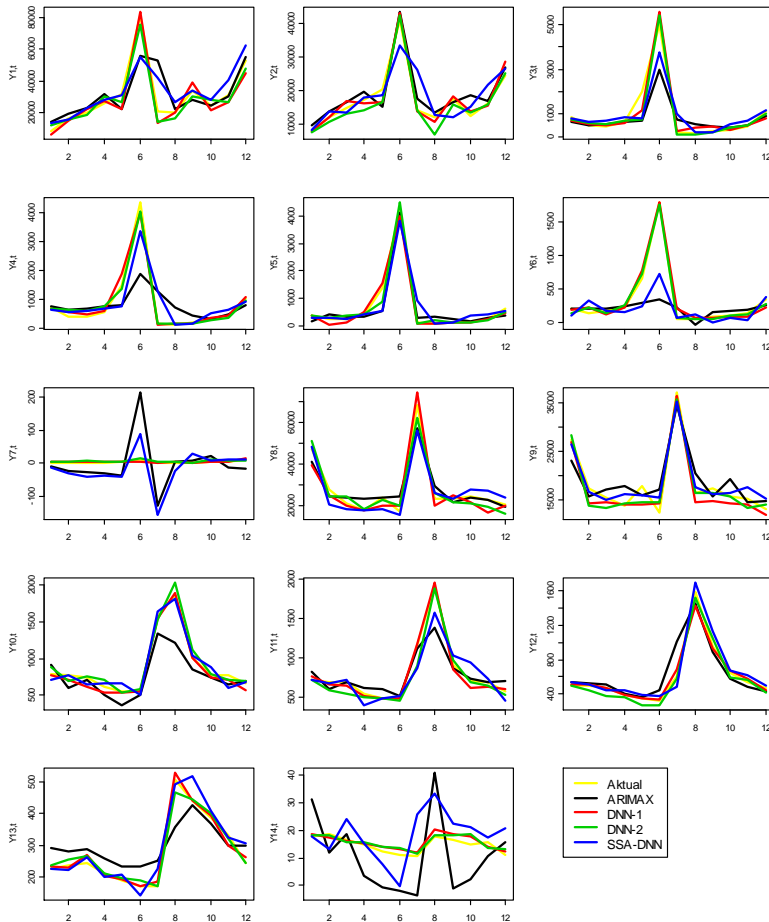
Metode DNN-1 dan DNN-2 menunjukkan konsistensi ramalan yang paling stabil dari 3 periode hingga 12 periode dan secara umum lebih baik dibanding ARIMAX dan SSA-DNN. Sedangkan ARIMAX dan SSA-DNN cenderung baik pada 3 periode.



Gambar 4.57. Perbandingan RMSEP ARIMAX, DNN-1, DNN-2 dan SSA-DNN Data *Outflow* Menurut Periode Ramalan

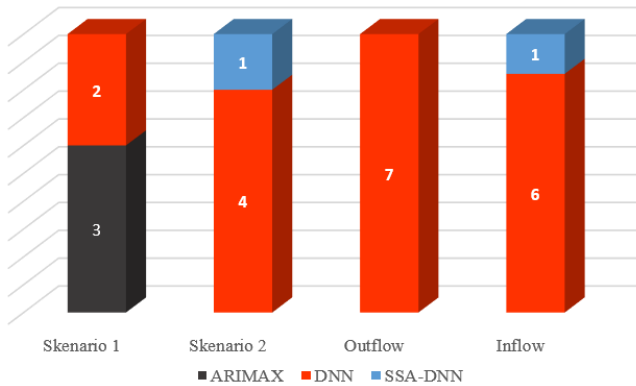
Hasil perbandingan RMSEP menurut periode pada data *inflow* menunjukkan hasil yang hampir sama dengan *outflow*. SSA-DNN unggul pada *inflow* Rp 50.000,- sedangkan DNN-2 unggul pada *inflow* Rp 100.000,- sementara pada pecahan lain metode DNN-1 memiliki akurasi yang terbaik. Hasil ini secara konsisten ditunjukkan pada 6 periode hingga 12 periode.

Berikut merupakan perbandingan antara data aktual dengan ramalan *testing* ARIMAX, DNN-1, DNN-2 dan SSA-DNN.



Gambar 4.58. Perbandingan *Testing* Aktual, ARIMAX, DNN-1, DNN-2 dan SSA-DNN

Hasil penentuan model terbaik keseluruhan pada kajian simulasi dan kajian terapan ditampilkan dalam ringkasan seperti pada Gambar 4.59.



Gambar 4.59. Ringkasan Hasil Pemilihan Model Terbaik

Hasil pemilihan model menunjukkan bahwa dari keseluruhan data yang digunakan DNN-1 dan DNN-2 merupakan model terbaik. DNN-2 dominan pada data simulasi dan beberapa data *outflow* serta 1 data *inflow*. Sedangkan DNN-1 adalah model dengan akurasi paling tinggi dibandingkan model lain pada kajian terapan yaitu 9 dari 14 pecahan. ARIMAX merupakan model terbaik untuk 3 dari 5 data simulasi dengan pola linier. Sementara itu SSA-DNN hanya unggul pada 1 dari 14 pecahan dan 1 dari 10 data simulasi. Kurang optimalnya performa SSA-DNN diduga karena adanya ketidak-konsistenan tren yang berubah pada tahun 2007 dan 2010. Sedangkan tren dan musiman yang berhasil direkonstruksi SSA cenderung monoton seperti Gambar 4.51. Kemungkinan lain adalah penggunaan variabel pada *trend analysis* dan regresi *dummy* yang kurang sesuai dengan pola hasil rekonstruksi SSA ataupun pengelompokan *eigentriple* yang kurang optimal.

Hasil serupa pernah ditulis dalam penelitian Lago (2018) yang menyimpulkan bahwa secara umum model *machine learning* menghasilkan akurasi lebih baik dibanding metode statistika seperti ARIMAX. Lago juga menyatakan bahwa pada kasusnya, model *hybrid* tidak lebih baik dibandingkan model yang lebih sederhana. M3 Competition menyimpulkan bahwa model yang

lebih rumit tidak serta merta meningkatkan akurasi, Namun hasil penelitian ini menghasilkan kesimpulan yang berbeda dengan M4, bahwa dalam M4 Competition, model *hybrid* dan model kombinasi adalah model terbaik dibanding model ML, dan model sederhana. Ringkasan pola data dan model terbaik ditampilkan dalam Tabel 4.30 berdasarkan hasil *white test* dalam Lampiran 35 yaitu pengujian antara setiap *series* dengan lag sesuai lag ARIMAX.

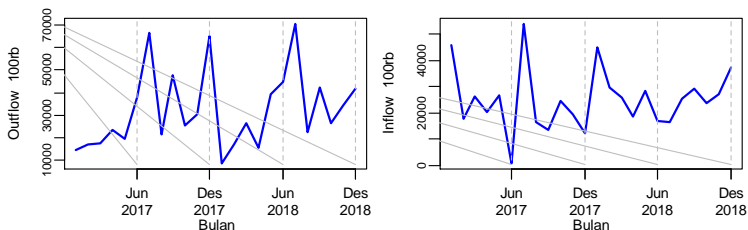
Tabel 4.30 Ringkasan Pola Data & Model Terbaik

Pola	Model Terbaik			Total
	ARIMAX	DNN	Hybrid	
Linier	0	4	0	4
Non-linier	0	9	1	10
Total	0	13	1	14

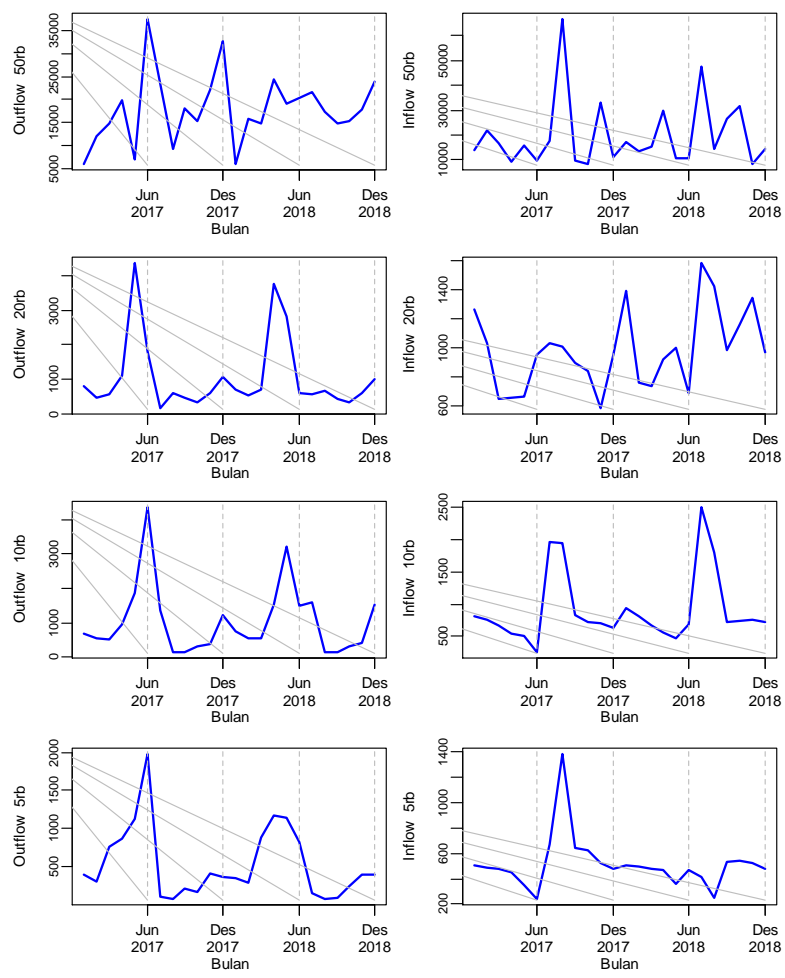
Tabel 4.30 menunjukkan bahwa model terbaik cenderung pada DNN untuk semua pola data, hal ini kemungkinan juga karena asumsi normalitas yang tidak terpenuhi pada data berpola linier.

4. 8 Peramalan Dengan Metode Terbaik

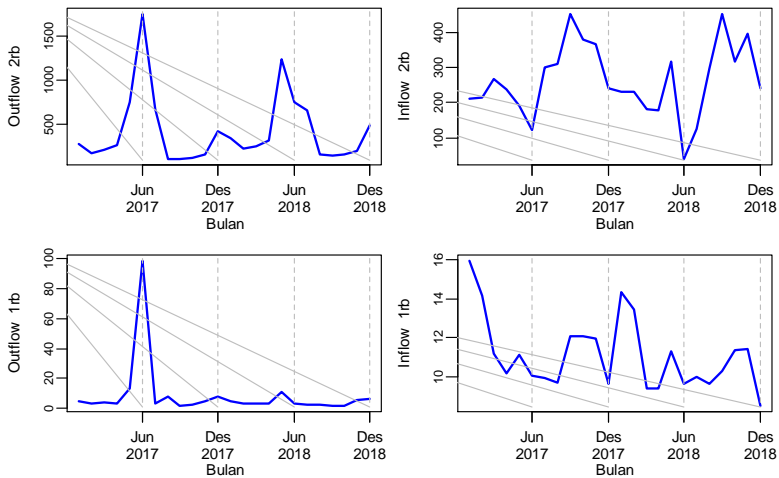
Peramalan hingga periode 2018 dilakukan menggunakan data *inflow* dan *outflow* 2003-2016. Metode yang digunakan adalah metode terbaik masing-masing pecahan yaitu metode DNN-2 untuk *outflow* Rp 100.000,-, Rp 10.000,- dan Rp 2.000,- serta *inflow* Rp 100.000,-. Metode SSA-DNN untuk pecahan *inflow* Rp 50.000,- sedangkan pecahan lain menggunakan metode DNN-1. Hasil ramalan ditampilkan dalam Gambar 4.60.



Gambar 4.60. Ramalan 24 Periode Kedepan Berdasarkan Model Terbaik



Gambar 4.60 (Lanjutan) Ramalan 24 Periode Kedepan Berdasarkan Model Terbaik



Gambar 4.60 (Lanjutan) Ramalan 24 Periode Kedepan Berdasarkan Model Terbaik

Hasil ramalan menunjukkan bahwa secara umum tren tidak terlalu terlihat dalam 24 periode ramalan. Namun efek variasi kalender masih terjadi dan ditunjukkan oleh puncak-puncak ramalan yang bersesuaian dengan bulan terjadinya Idul Fitri serta bulan sebelum atau sesudahnya. Efek variasi kalender paling terlihat pada *outflow* pecahan Rp 20.000,- hingga *outflow* Rp 1.000,- karena umumnya kebutuhan uang yang meningkat drastis pada Hari Raya Idul Fitri adalah pecahan tersebut. Sedangkan pada pecahan Rp 100.000,- dan Rp 50.000,- efek variasi kalender tidak sebesar pecahan lainnya. Selain itu, pada pecahan Rp 1.000,- baik *outflow* maupun *inflow*, ramalan cenderung terus menurun dan menuju nol. Hal ini kemungkinan disebabkan oleh penggunaan pecahan tersebut yang semakin jarang dan banyak tergantikan oleh pecahan Rp 2.000,-.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Analisis dan pembahasan yang dilakukan menunjukkan hasil yang mendukung beberapa kesimpulan sebagai berikut :

1. Pada kajian simulasi, ARIMAX tidak berbeda dengan DNN dan SSA-DNN untuk peramalan data yang mengandung *noise* linier baik pada ramalan periode pendek maupun ramalan periode yang lebih panjang. Pada data dengan *noise* non-linier performa ARIMAX tidak optimum. Sedangkan pada data *outflow* dan *inflow*, ARIMAX memiliki akurasi yang lebih rendah dibandingkan DNN dan SSA-DNN pada semua pecahan.
2. DNN menghasilkan akurasi terbaik dibandingkan model lain pada data simulasi dengan pola non-linier, yaitu DNN yang menggunakan input lag mengikuti lag AR dari model ARIMAX untuk ramalan dengan periode yang lebih panjang dan DNN yang menggunakan input lag PACF signifikan dari data untuk periode ramalan yang lebih pendek. Sedangkan pada data *outflow* dan *inflow* DNN unggul pada 13 dari 14 pecahan yang dimodelkan, baik *training* maupun *testing* dibandingkan dengan ARIMAX dan SSA-DNN. 9 diantaranya menggunakan input lag PACF yang signifikan dari data serta variabel *dummy*, sedangkan 4 lainnya menggunakan input lag AR dari ARIMAX serta variabel *dummy*. Selain itu, DNN dapat menangkap efek variasi kalender dengan sangat baik dengan input variabel *dummy*.
3. SSA-DNN hampir sama baiknya dengan DNN ketika meramalkan data simulasi dengan pola non-linier. Namun pada data *outflow* dan *inflow* SSA-DNN memiliki akurasi lebih baik dibandingkan ARIMAX dan DNN hanya pada 1

pecahan yaitu *inflow* Rp 50.000,- walaupun secara umum, SSA-DNN sudah mampu menangkap efek variasi kalender namun DNN masih lebih unggul.

4. Secara umum, DNN dan SSA-DNN dapat diterapkan pada data yang mengandung pola linier maupun non-linier. Namun akurasi terbaik untuk peramalan *outflow* dan *inflow* dihasilkan oleh model DNN.
5. Ramalan 24 periode hingga Desember 2018 menunjukan efek variasi kalender yang mencolok pada pecahan Rp 20.000,- kebawah terutama untuk *outflow*. Ramalan pecahan Rp 1.000,- baik *outflow* maupun *inflow* cenderung memiliki tren turun menuju nol karena mulai jarang digunakan.

5.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah untuk melakukan pemilihan input DNN yang lebih komprehensif dengan *cross validation*. Pemodelan hasil rekonstruksi SSA dengan model yang disesuaikan untuk setiap hasil rekonstruksi juga perlu dicobakan untuk mengetahui performa optimum dari SSA-DNN. Peramalan interval dari model DNN dan *hybrid* DNN dapat dilakukan menggunakan *bootstrap* sebagai informasi tambahan untuk membandingkan kebaikan antar model dan prediksi periode kedepan. Selain itu deteksi *outlier* dan penggunaan model GARCH pada ARIMAX dapat dilakukan untuk pemodelan yang lebih baik dalam mengatasi asumsi yang tidak terpenuhi. Kajian simulasi dengan replikasi yang lebih banyak dapat membantu memperoleh kesimpulan yang lebih reliabel dengan mencobakan berbagai *preprocessing*, kombinasi input, kombinasi neuron dan beberapa fungsi aktivasi berbeda terutama untuk model DNN dan *hybrid* DNN.

DAFTAR PUSTAKA

- Ahmad, I. S., Setiawan, Suhartono, & Masun, N. H. (2015). Forecasting of Monthly Inflow and Outflow Currency Using Time Series Regression and ARIMAX: The Idul Fitri Effect. *AIP Conference Proceedings* (hal. 1-8). American Institute of Physics.
- Alzahrani, A., Shamsi, P., Dagli, C., Ferdowsi, M. (2017) Solar Irradiance Forecasting Using Deep Neural Network. *Procedia Computer Science*, 114, 304-313.
- Ambarini, L. (2015). *Ekonomi Moneter*. Bogor: Penerbit IN MEDIA.
- Anders, U., & Korn, O. (1999). Model Selection in Neural Networks. *Neural Networks*, 12, 309-323.
- Bai, Y., Chen, Z., Xie, J., Li, C. (2016). Daily Reservoir Inflow Forecasting Using Multiscale Deep Feature Learning with Hybrid Models. *Journal of Hydrology*, 532, 193-206
- Bank Indonesia. (2011, Juli). Peredaran Uang. *Gera Info*, hal. 1-8.
- Bank Indonesia. (2018, Februari 17). *Metadata SSKI*. Diambil kembali dari Bank Indonesia Official Website: <http://www.bi.go.id/id/statistik/metadata/SSKI/Documents/12Metadata%20Uang%20Kartal%20yang%20Diedarkan.pdf>
- Bowerman, B. L., & O'Connell, R. T. (1993). *Forecasting and Time Series*. Belmont: Wadsworth Publishing Company.
- Busseti, E. (2012). *Deep learning for Time Series Modelling*. Stanford: Stanford University
- Chen, Z., & Yang, Y. (2004). Assessing forecast accuracy measures. Preprint Series, 2010, 2004-10.
- Crone, S. F., & Kourentzes, N. (2009). Input-variable Specification for Neural Network - an Analysis of Forecasting Low and High Time Series Frequency. *International Joint Conference on Neural Network*, (hal. 14-19).

- Cryer, J. D., & Chan, K. S. (2008). *Time Series Analysis with Applications in R*. New York: Springer Science.
- Daniel, W. W. (1989). *Statistik Nonparametrik Terapan*. (A. T. W, Penerj.) Jakarta: PT. Gramedia.
- Daniel, W. (1989). *Statistik Nonparametrik Terapan (Terjemahan Widodo)*. Jakarta: PT. Gramedia Pustaka Utama.
- Golyandina, N., & Zhigljavsky, A. (2013). *Singular Spectrum Analysis for Time Series*. Heidelberg: Springer-Verlag Berlin.
- Golyandina, N., Nekrutkin, V., & Zhigljavsky, A. A. (2001). *Analysis of Time Series Structure : SSA and Related Techniques*. Florida: Chapman & Hall / CRC.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge: MIT Press.
- Gorr, W. L. (1994). Research prospective on neural network forecasting. *International Journal of Forecasting*, 10(1), 1-4.
- Guler, H. & Talasli A.. (2010). *Modelling the Daily Currency in Circulation in Turkey*. Ankara : Turkey Central Bank
- Hamilton, J. D. (1994). *Time Series Analysis*. New Jersey: Princeton University Press.
- He, W.(2017). Load Forecasting via Deep Neural Networks. *Procedia Computer Science*, 122, 308-314
- Kumar, U. & Jain, V.K. (2010). Time Series Models (Grey-Markov, Grey Model with Rolling Mechanism and Singular Spectrum Analysis) to forecast energy consumption in India. *Energy*, 35, 1709-1716
- Lago, J., Ridder, F.D., Schutter, B.D. (2018). Forecasting Spot Electricity Prices: Deep Learning Approach and Empirical Comparison of Tradotional Algorithms. *Applied Energy*, 221, 386-405
- Lasheras, F.S., Vilan, J.A.V., Nieto, P.J.G., & Diaz, J.J.d.C. (2010). The use of design of experiments to improve a neural

- network model in order to predict the thickness of the chromium layer in a hard chromium plating process. *Mathematical and Computer Modelling*, 52, 1169-1176.
- Lee, T.H., White, H., & Granger, C.W.J. (1993). Testing for neglected nonlinearity in time series models. *Journal of Econometrics* 56, 269-290.
- Liu, H., Mi, X., & Li, Y. (2018). Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. *Energy Conversion and Management*, 159, 54-64
- Makridakis, S., Hibon, M. (2000) The M3-Competition: Results, Conclusions, and Implications. *International Journal of Forecasting*, 16, 451-476
- Makridakis, S., Spiliotis, E., Assimakopoulos .(2018). Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. *PloS ONE*, 13
- Makridakis, S., Spiliotis, E., Assimakopoulos .(2018). The M4 Competition: Results, Findings, Conclusions and way forward. *International Journal of Forecasting*.
- Patterson, J., & Gibson, A. (2017). *Deep Learning: A Practitioner's Approach*. California: O'Reilly Media.
- Rachmawati, N. I. (2005). *Peramalan inflow dan outflow Uang Kartal Bank Indonesiadi Wilayah Purwokerto*. Surabaya: ITS
- Salehah, N.A., (2017). *Penerapan Model Hybrid ARIMAX-Quantile Regression untuk Peramalan Inflow dan Outflow Pecahan Uang Kartal di Jawa Timur*. Surabaya:ITS
- Saputri, P. D. (2017). *Aplikasi Model Hybrid Quantile Regression Neural Network Pada Peramalan Pecahan Inflow dan Outflow Uang Kartal Di Indonesia*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Setyowati, E. (2017). *Model Hybrid Singular Spectrum Analysis Dan Neural Network Untuk Peramalan Nilai Pecahan*

Inflow Dan Outflow Uang Kartal Di Indonesia. Surabaya: ITS.

- Suhartono. (2007). *Feedforward Neural Network untuk Pemodelan Runtun Waktu*. Yogyakarta: Disertasi Doktorat, Universitas Gajah Mada.
- Thinsungneon, T., Kerdprasop, K., & Kerdprasop, N. (2017). A Deep Learning of Time Series for Efficient Analysis. *International Journal of Future Computer and Communication*, 6, 123-127.
- Vautard, R. & Ghil, M. (1989). Singular Spectrum Analysis in Nonlinear Dynamics, with Applications to Paleoclimatic Time Series. *Physica D: Nonlinear Phenomena*, 35, 395-424.
- Wei, W. W. (2006). *Time Series Analysis Univariate and Multivariate Methods* (2nd ed.). United States of America: Pearson Education, Inc.
- Yu, C., Li, Y., & Zhang, M. (2017). Comparative study on three new hybrid models using Elman Neural Network and Empirical Mode Decomposition based technologies improved by Singular Spectrum Analysis for hour-ahead wind speed forecasting. *Energy Conversion and Management*, 147, 76-85.
- Zhang, Q., Wang, B.-D., He, B., & Peng, Y. (2011). *Singular Spectrum Analysis and ARIMA Hybrid Model for Annual Runoff Forecasting*. China: Springer.
- Zhang, X., Wang, J., & Zhang, K. (2017). Short-term Electric Load Forecasting Based on Singular Spectrum Analysis and Support Vector Machine Optimized by Cuckoo Search Algorithm. *Electric Power Systems Research*, 146, 270-285.
- Zubaidi, S.L., Dooley, J., Alkhaddar, R.M., Abdellatif, M., Al-Bugharbee, H., OrtegaMartorell, S., A (2018) Novel Approach for Predicting Monthly Water Demand by Combining Singular Spectrum Analysis with Neural Networks. *Journal of Hydrology*

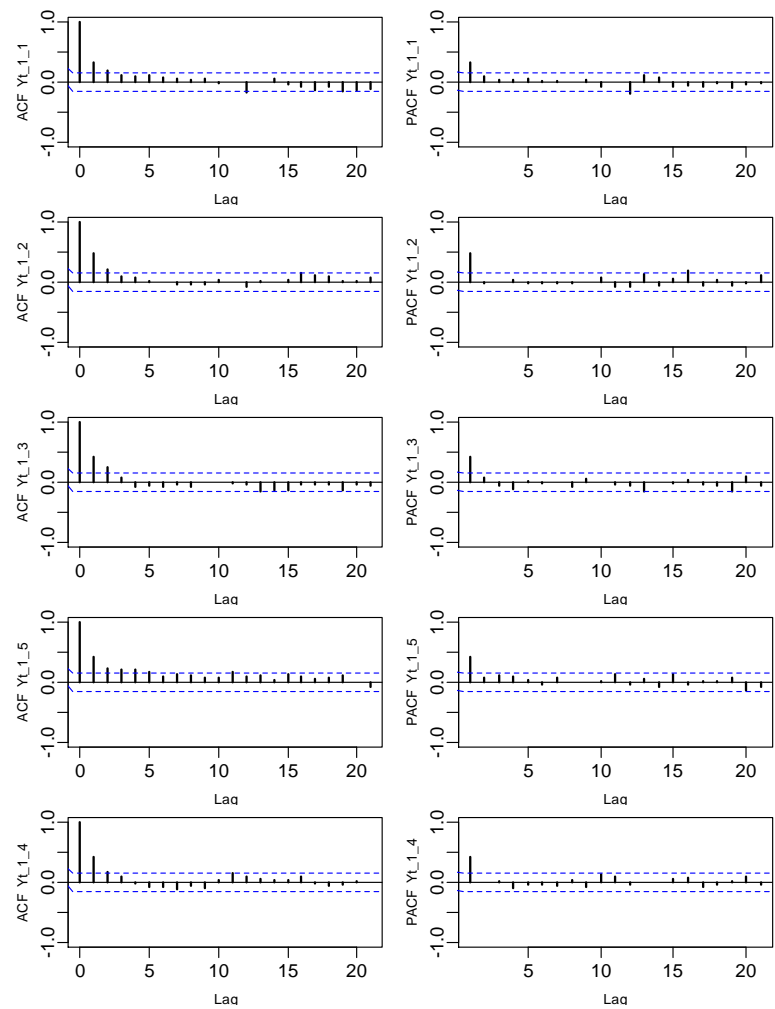
LAMPIRAN

Lampiran 1. Data Inflow & Outflow

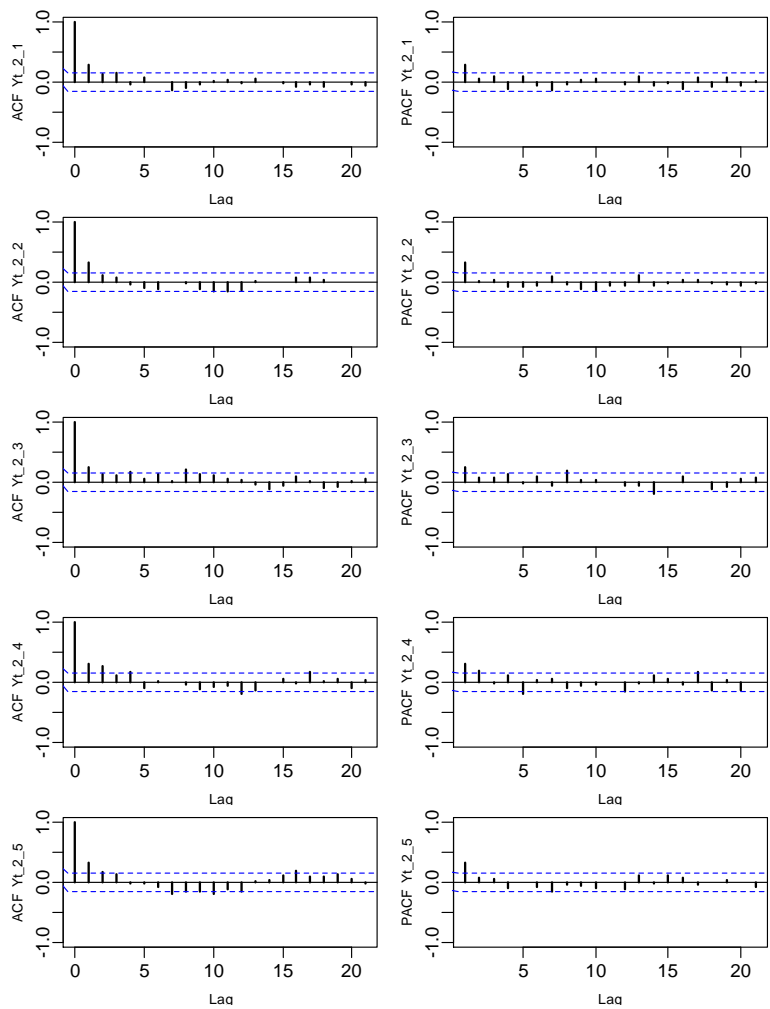
<i>Outflow (miliar Rp)</i>						
Tahun	Bulan	<i>Pecahan</i>				
		100.000	50.000	20.000	...	1.000
2003	Januari	5,275,028	8,146,788	1,192,291	...	123,186
2003	Februari	4,274,460	8,685,145	1,092,891	...	101,478
⋮	⋮	⋮	⋮	⋮		⋮
2015	November	25,280,419	14,840,341	498,633	...	3,199
2015	Desember	53,484,049	29,327,246	1,049,066	...	4,518
2016	Januari	8,302,630	7,867,459	842,464	...	2,382
2016	Februari	16,450,722	12,009,425	505,229		2,283
⋮	⋮	⋮	⋮	⋮		⋮
2016	November	30,681,523	16,104,669	474,291	...	2,844
2016	Desember	52,448,688	24,550,124	1,146,138	...	13,145

<i>Inflow (miliar Rp)</i>						
Tahun	Bulan	<i>Pecahan</i>				
		100.000	50.000	20.000	...	1.000
2003	Januari	7,666,245	12,424,344	2,096,296	...	278,244
2003	Februari	5,308,464	8,616,926	1,399,736	...	187,763
⋮	⋮	⋮	⋮	⋮		⋮
2015	November	18,856,685	14,425,381	663,971	...	18,625
2015	Desember	13,436,752	10,632,061	614,987	...	15,893
2016	Januari	47,141,781	27,297,007	797,266	...	17,933
2016	Februari	27,622,141	17,300,484	769,849	...	18,499
⋮	⋮	⋮	⋮	⋮		⋮
2016	November	22,409,779	15,030,069	778,572	...	15,547
2016	Desember	20,162,623	13,001,197	20,162,623	...	478,174

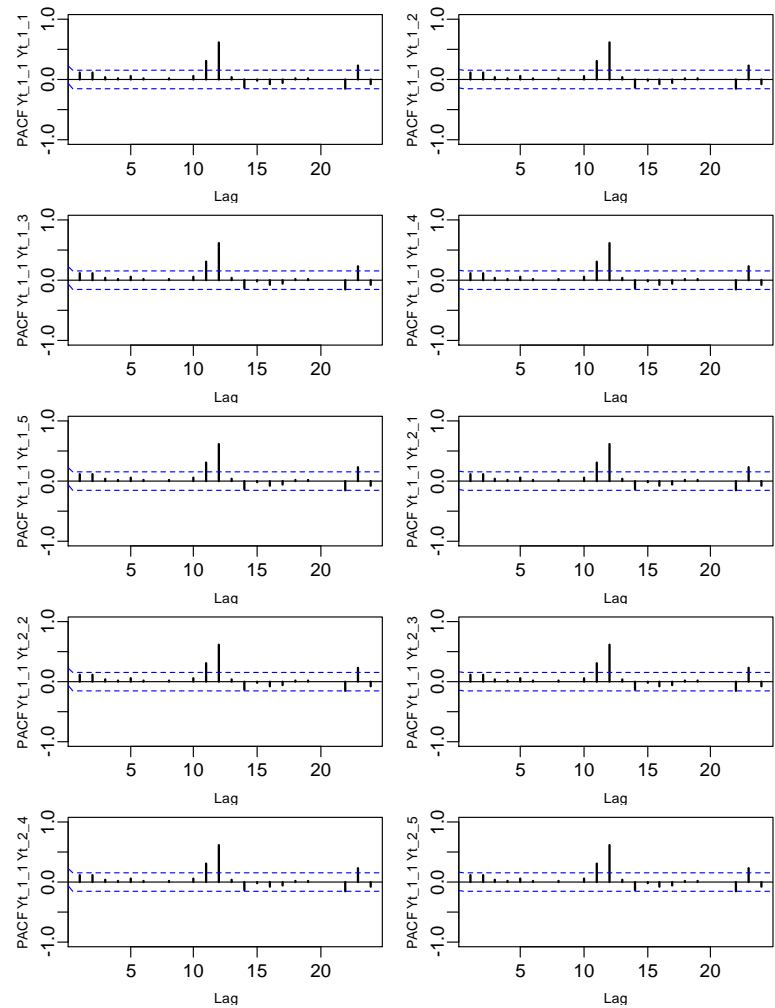
Lampiran 2. ACF dan PACF Residual TSR Skenario 1 Simulasi



Lampiran 3. ACF dan PACF Residual TSR Skenario 2 Simulasi



Lampiran 4. PACF Data Simulasi untuk input DNN-1



Lampiran 5. Akurasi Data Simulasi

Data	ARIMAX				DNN-1			
	RMSE	sMAPE	RMSEP	sMAPEP	RMSE	sMAPE	RMSEP	sMAPEP
Y1_1,t	0.955	3.414	1.027	2.113	1.014	3.490	1.431	2.772
Y1_2,t	0.902	3.447	0.988	2.517	0.906	2.962	1.185	2.912
Y1_3,t	0.998	3.897	1.710	4.845	1.098	3.860	1.905	5.602
Y1_4,t	0.975	3.762	0.879	2.690	0.790	2.617	1.650	4.492
Y1_5,t	0.996	3.536	0.729	1.957	1.087	3.725	1.641	2.699
Y2_1,t	2.824	11.363	3.876	9.112	1.286	3.224	3.153	9.015
Y2_2,t	2.605	10.185	4.125	11.267	2.723	9.793	4.182	10.920
Y2_3,t	3.003	12.025	2.273	5.983	3.034	10.779	2.152	5.697
Y2_4,t	2.802	11.790	3.646	10.363	2.928	10.394	3.165	9.422
Y2_5,t	2.869	11.490	3.934	10.266	2.967	10.073	3.593	10.495

Data	DNN-2				SSA-DNN			
	RMSE	sMAPE	RMSEP	sMAPEP	RMSE	sMAPE	RMSEP	sMAPEP
Y1_1,t	0.883	3.052	0.960	2.192	1.860	4.328	1.686	4.909
Y1_2,t	0.685	2.059	2.886	5.354	2.057	4.431	1.435	3.828
Y1_3,t	0.890	3.494	1.242	3.342	2.027	5.186	1.564	4.484
Y1_4,t	0.757	2.924	1.362	3.463	2.373	7.068	1.217	2.722
Y1_5,t	0.790	2.590	1.280	3.327	1.809	4.598	1.227	3.198
Y2_1,t	2.811	11.438	3.496	9.774	2.539	7.764	3.572	9.709
Y2_2,t	1.882	5.517	3.362	7.785	2.461	7.659	2.924	6.974
Y2_3,t	2.822	9.422	1.972	5.454	3.412	12.126	2.147	6.327
Y2_4,t	2.440	8.849	3.000	8.314	1.375	4.110	3.291	9.970
Y2_5,t	2.701	8.698	3.239	9.086	1.385	3.305	3.509	9.088

Paired T-test

Perbandingan	Skenario 1			Skenario 2		
	T	P	Kesimpulan	T	P	Kesimpulan
ARIMAX vs DNN	-0.740	0.499	tidak berbeda signifikan	7.500	0.002	DNN
ARIMAX vs SSA-DNN	-2.630	0.058	tidak berbeda signifikan	2.590	0.061	tidak berbeda signifikan
DNN vs SSA-DNN	-1.430	0.227	tidak berbeda signifikan	-0.950	0.396	tidak berbeda signifikan

Lampiran 6. Estimasi Parameter ARIMAX *Outflow* Rp.100.000,-

Variabel	Estimasi	SE	T	P
MA2	-0.201	0.092	-2.190	0.031
AR10	-0.185	0.059	-3.140	0.002
AR12	0.815	0.062	13.190	<.0001
NUM1	280.607	42.641	6.580	<.0001
NUM2	18941.900	5659.200	3.350	0.001
NUM3	12308.400	7282.100	1.690	0.093
NUM4	-147.597	66.287	-2.230	0.028
NUM5	-147.347	74.939	-1.970	0.051
NUM6	-17528.900	6167.300	-2.840	0.005
NUM7	-16331.700	6188.900	-2.640	0.009
NUM8	-18424.900	6070.500	-3.040	0.003
NUM9	-15692.500	6102.400	-2.570	0.011
NUM10	-15242.300	6060.100	-2.520	0.013
NUM11	-12726.100	6098.200	-2.090	0.039
NUM12	-11210.600	6141.300	-1.830	0.070
NUM13	-16193.900	6129.600	-2.640	0.009
NUM14	-12917.800	6166.900	-2.090	0.038
NUM15	-15518.600	6262.900	-2.480	0.015
NUM16	-23686.000	6353.400	-3.730	0.000
NUM17	-8195.500	6190.600	-1.320	0.188
NUM18	18198.000	2611.900	6.970	<.0001
NUM19	3354.600	2037.900	1.650	0.102
NUM20	-4043.800	2157.800	-1.870	0.063
NUM21	1272.700	1736.200	0.730	0.465
NUM22	2815.700	2155.800	1.310	0.194
NUM23	11505.200	1843.100	6.240	<.0001
NUM24	16131.400	2444.700	6.600	<.0001
NUM25	17060.800	2106.800	8.100	<.0001

Lampiran 7. Estimasi Parameter ARIMAX *Outflow* Rp.50.000,-

Paramater	Estimasi	SE	T	P	Lag	Variabel
AR1,1	0.354	0.085	4.160	<.0001	12.000	y
AR1,2	-0.262	0.086	-3.050	0.003	13.000	y
AR1,3	0.213	0.092	2.320	0.022	23.000	y
AR1,4	-0.226	0.092	-2.460	0.015	26.000	y
NUM1	116.192	22.710	5.120	<.0001	0.000	t
NUM2	13306.300	3003.100	4.430	<.0001	0.000	d1
NUM3	8142.600	3802.200	2.140	0.034	0.000	d2
NUM4	-78.956	38.228	-2.070	0.041	0.000	td1
NUM5	-109.263	39.517	-2.760	0.007	0.000	td2
NUM6	-6254.000	3002.300	-2.080	0.039	0.000	M1
NUM7	-4155.800	3035.000	-1.370	0.173	0.000	M2
NUM8	-1575.000	3047.400	-0.520	0.606	0.000	M3
NUM9	-1458.200	3052.400	-0.480	0.634	0.000	M4
NUM10	-848.374	3060.800	-0.280	0.782	0.000	M5
NUM11	1130.600	3077.700	0.370	0.714	0.000	M6
NUM12	979.407	3098.900	0.320	0.753	0.000	M7
NUM13	-3020.700	3109.800	-0.970	0.333	0.000	M8
NUM14	-3064.400	3119.400	-0.980	0.328	0.000	M9
NUM15	-2454.300	3146.900	-0.780	0.437	0.000	M10
NUM16	-2864.500	3138.300	-0.910	0.363	0.000	M11
NUM17	6927.600	3095.200	2.240	0.027	0.000	M12
NUM18	20510.500	2376.200	8.630	<.0001	0.000	L1b
NUM19	5332.600	1783.000	2.990	0.003	0.000	L2b
NUM20	-2839.200	1868.800	-1.520	0.131	0.000	L3b
NUM21	63.683	1490.000	0.040	0.966	0.000	L4b
NUM22	-2427.100	2025.900	-1.200	0.233	0.000	L1
NUM23	12848.200	1684.000	7.630	<.0001	0.000	L2
NUM24	14847.000	2065.200	7.190	<.0001	0.000	L3
NUM25	15867.900	1756.700	9.030	<.0001	0.000	L4

Lampiran 8. Estimasi Parameter ARIMAX *Outflow* Rp.20.000,-

Variabel	Estimasi	SE	T	P-value
t	2.249	2.217	1.015	0.312
D1	1295.346	294.222	4.403	0.000
D2	561.854	359.751	1.562	0.121
tD1	-11.172	3.789	-2.949	0.004
tD2	-7.219	3.763	-1.919	0.057
M1,t	299.208	288.697	1.036	0.302
M2,t	127.355	289.512	0.440	0.661
M3,t	200.432	290.334	0.690	0.491
M4,t	293.375	291.162	1.008	0.316
M5,t	350.159	291.998	1.199	0.233
M6,t	578.352	293.603	1.970	0.051
M7,t	577.157	295.813	1.951	0.053
M8,t	189.132	296.095	0.639	0.524
M9,t	84.350	297.846	0.283	0.777
M10,t	23.039	300.795	0.077	0.939
M11,t	131.023	299.282	0.438	0.662
M12,t	556.742	298.027	1.868	0.064
L1,t-1	2039.463	219.446	9.294	0.000
L2,t-1	1379.705	155.346	8.881	0.000
L3,t-1	1463.272	178.797	8.184	0.000
L4,t-1	459.480	155.427	2.956	0.004
L1,t	-205.254	218.957	-0.937	0.350
L2,t	469.441	154.981	3.029	0.003
L3,t	1192.603	179.466	6.645	0.000
L4,t	1813.529	155.627	11.653	< 2e-16

Lampiran 9. Estimasi Parameter ARIMAX *Outflow* Rp.10.000,-

Variabel	Estimasi	SE	T	P
t	4.314	2.388	1.807	0.073
d1	456.141	316.911	1.439	0.152
d2	164.168	387.494	0.424	0.673
td1	-5.793	4.081	-1.420	0.158
td2	-2.140	4.053	-0.528	0.598
m_1	78.270	310.960	0.252	0.802
m_2	-18.085	311.838	-0.058	0.954
m_3	9.368	312.723	0.030	0.976
m_4	65.713	313.616	0.210	0.834
m_5	98.000	314.516	0.312	0.756
m_6	254.651	316.245	0.805	0.422
m_7	367.823	318.625	1.154	0.250
m_8	15.312	318.929	0.048	0.962
m_9	-286.033	320.815	-0.892	0.374
m_10	-399.977	323.991	-1.235	0.219
m_11	-249.246	322.362	-0.773	0.441
m_12	100.502	321.009	0.313	0.755
Lb_1	947.958	236.369	4.011	0.000
Lb_2	1473.464	167.326	8.806	0.000
Lb_3	1247.497	192.585	6.478	0.000
Lb_4	434.054	167.413	2.593	0.011
L_1	204.532	235.842	0.867	0.387
L_2	400.878	166.933	2.401	0.018
L_3	960.395	193.306	4.968	0.000
L_4	1906.673	167.628	11.374	< 2e-16

Lampiran 10. Estimasi Parameter ARIMAX *Outflow* Rp.5.000,-

Variabel	Estimasi	SE	T	P
AR1	-0.166	0.059	-2.790	0.006
AR12	0.410	0.064	6.390	<.0001
AR13	0.694	0.075	9.210	<.0001
t	2.502	1.527	1.640	0.104
d1	268.101	185.293	1.450	0.150
d2	182.228	214.059	0.850	0.396
td1	-3.199	2.267	-1.410	0.161
td2	-2.870	2.197	-1.310	0.194
M1	-88.068	194.707	-0.450	0.652
M2	-80.477	196.126	-0.410	0.682
M3	-44.510	196.545	-0.230	0.821
M4	42.493	196.555	0.220	0.829
M5	122.741	195.848	0.630	0.532
M6	211.644	194.794	1.090	0.279
M7	317.423	195.252	1.630	0.107
M8	16.947	196.273	0.090	0.931
M9	-214.244	197.777	-1.080	0.281
M10	-319.574	201.698	-1.580	0.116
M11	-241.258	200.107	-1.210	0.230
M12	-65.771	197.618	-0.330	0.740
L1b	1620.700	205.359	7.890	<.0001
L2b	1175.500	170.762	6.880	<.0001
L3b	804.894	176.681	4.560	<.0001
L4b	359.865	116.364	3.090	0.002
L1	-104.891	140.805	-0.740	0.458
L2	164.538	143.585	1.150	0.254
L3	521.237	186.554	2.790	0.006
L4	903.873	162.190	5.570	<.0001

Lampiran 11. Estimasi Parameter ARIMAX *Outflow* Rp.2.000,-

Variabel	Estimasi	SE	T	P
t	1.664	0.643	2.586	0.012
d1	NA	NA	NA	NA
d2	2.882	55.846	0.052	0.959
td1	NA	NA	NA	NA
td2	-0.790	4.222	-0.187	0.852
M1.t.	72.066	44.848	1.607	0.114
M2.t.	66.288	45.167	1.468	0.148
M3.t.	71.590	45.500	1.573	0.121
M4.t.	96.327	45.848	2.101	0.040
M5.t.	155.208	46.210	3.359	0.001
M6.t.	201.015	49.691	4.045	0.000
M7.t.	76.256	53.718	1.420	0.161
M8.t.	-182.375	53.410	-3.415	0.001
M9.t.	2.944	47.750	0.062	0.951
M10.t.	13.882	46.420	0.299	0.766
M11.t.	30.622	46.778	0.655	0.515
M12.t.	112.395	47.167	2.383	0.021
L1.t.1.	NA	NA	NA	NA
L2.t.1.	1480.553	70.557	20.984	<0.001
L3.t.1.	772.716	57.327	13.479	<0.001
L4.t.1.	182.969	67.686	2.703	0.009
L1.t.	NA	NA	NA	NA
L2.t.	230.804	68.464	3.371	0.001
L3.t.	781.177	56.503	13.825	<0.001
L4.t.	1540.713	69.603	22.136	<0.001

Lampiran 12. Estimasi Parameter ARIMAX *Outflow* Rp.1.000,-

Variabel	Estimasi	SE	T	P
AR1	0.458	0.081	5.660	<.0001
AR11	0.256	0.085	3.000	0.003
AR13	-0.216	0.084	-2.590	0.011
t	-0.795	0.473	-1.680	0.095
d1	37.036	62.400	0.590	0.554
d2	244.680	79.340	3.080	0.003
td1	1.228	0.777	1.580	0.117
td2	-2.854	0.824	-3.460	0.001
M1	107.097	60.354	1.770	0.078
M2	93.021	60.371	1.540	0.126
M3	94.717	60.489	1.570	0.120
M4	94.931	60.643	1.570	0.120
M5	95.583	60.809	1.570	0.119
M6	97.042	61.038	1.590	0.114
M7	84.527	61.327	1.380	0.171
M8	127.486	61.458	2.070	0.040
M9	139.533	61.613	2.260	0.025
M10	147.936	61.951	2.390	0.018
M11	108.075	61.900	1.750	0.083
M12	112.924	61.751	1.830	0.070
L1b	276.083	25.140	10.980	<.0001
L2b	104.666	17.632	5.940	<.0001
L3b	24.833	19.705	1.260	0.210
L4b	49.631	16.930	2.930	0.004
L1	-65.621	23.808	-2.760	0.007
L2	113.617	17.378	6.540	<.0001
L3	136.946	19.677	6.960	<.0001
L4	248.488	18.071	13.750	<.0001

Lampiran 13. Estimasi Parameter ARIMAX *Inflow* Rp.100.000,-

Parameter	Estimasi	SE	T	P	Lag	Variabel
MA1,1	0.203	0.081	2.500	0.014	1	y
MA1,2	-0.534	0.108	-4.960	<.0001	12	y
NUM1	256.364	34.104	7.520	<.0001	0	t
NUM2	19230.500	4431.100	4.340	<.0001	0	d1
NUM3	11120.400	5935.500	1.870	0.063	0	d2
NUM4	-175.257	50.335	-3.480	0.001	0	td1
NUM5	-136.228	60.517	-2.250	0.026	0	td2
NUM6	-6456.700	4591.800	-1.410	0.162	0	M1
NUM7	-14539.300	4609.700	-3.150	0.002	0	M2
NUM8	-15531.000	4623.100	-3.360	0.001	0	M3
NUM9	-15871.200	4631.800	-3.430	0.001	0	M4
NUM10	-16037.500	4641.400	-3.460	0.001	0	M5
NUM11	-16255.700	4648.900	-3.500	0.001	0	M6
NUM12	-13395.600	4679.500	-2.860	0.005	0	M7
NUM13	-11569.500	4688.900	-2.470	0.015	0	M8
NUM14	-16290.500	4700.000	-3.470	0.001	0	M9
NUM15	-15425.200	4711.900	-3.270	0.001	0	M10
NUM16	-18088.400	4774.600	-3.790	0.000	0	M11
NUM17	-20726.600	4761.300	-4.350	<.0001	0	M12
NUM18	19767.000	3380.700	5.850	<.0001	0	L1
NUM19	11382.600	2423.500	4.700	<.0001	0	L2
NUM20	11380.700	2676.500	4.250	<.0001	0	L3
NUM21	-3016.500	2140.300	-1.410	0.161	0	L4
NUM22	2824.200	3042.600	0.930	0.355	0	L1a
NUM23	2570.900	2333.700	1.100	0.273	0	L2a
NUM24	4453.200	2821.700	1.580	0.117	0	L3a
NUM25	16906.900	2511.500	6.730	<.0001	0	L4a

Lampiran 14. Estimasi Parameter ARIMAX *Inflow* Rp.50.000,-

Parameter	Estimasi	SE	T	P	Lag	Variabel
MA1,1	0.221	0.093	2.360	0.020	16	y
MA1,2	0.195	0.094	2.070	0.041	18	y
MA1,3	0.379	0.106	3.580	0.001	24	y
MA1,4	0.205	0.101	2.040	0.043	25	y
NUM1	109.612	19.734	5.550	<.0001	0	t
NUM2	11543.500	2554.300	4.520	<.0001	0	d1
NUM3	4823.700	3228.200	1.490	0.138	0	d2
NUM4	-24.180	31.983	-0.760	0.451	0	td1
NUM5	-70.187	33.430	-2.100	0.038	0	td2
NUM6	5070.100	2494.200	2.030	0.044	0	M1
NUM7	-630.604	2513.100	-0.250	0.802	0	M2
NUM8	-1130.600	2512.300	-0.450	0.654	0	M3
NUM9	-1122.800	2516.500	-0.450	0.656	0	M4
NUM10	-1960.200	2531.500	-0.770	0.440	0	M5
NUM11	-1877.400	2539.100	-0.740	0.461	0	M6
NUM12	-70.519	2555.800	-0.030	0.978	0	M7
NUM13	231.456	2573.200	0.090	0.929	0	M8
NUM14	-1746.200	2576.100	-0.680	0.499	0	M9
NUM15	-464.257	2572.900	-0.180	0.857	0	M10
NUM16	-3262.800	2575.100	-1.270	0.208	0	M11
NUM17	-4342.500	2574.200	-1.690	0.094	0	M12
NUM18	14964.900	1605.800	9.320	<.0001	0	L1
NUM19	10575.300	1137.800	9.290	<.0001	0	L2
NUM20	5060.600	1351.600	3.740	0.000	0	L3
NUM21	-3479.100	1096.700	-3.170	0.002	0	L4
NUM22	2672.900	1640.800	1.630	0.106	0	L1a
NUM23	3329.500	1118.500	2.980	0.004	0	L2a
NUM24	5725.500	1358.600	4.210	<.0001	0	L3a
NUM25	13086.500	1110.800	11.780	<.0001	0	L4a

Lampiran 15. Estimasi Parameter ARIMAX *Inflow* Rp.20.000,-

Parameter	Estimasi	SE	T	P	Lag	Variabel
MA1,1	-0.510	0.084	-6.040	<.0001	3	y
MA1,2	-0.294	0.080	-3.680	0.000	11	y
MA1,3	-0.380	0.078	-4.880	<.0001	12	y
MA1,4	0.305	0.082	3.700	0.000	20	y
AR1,1	0.219	0.094	2.320	0.022	1	y
AR1,2	0.250	0.095	2.630	0.010	2	y
AR1,3	0.205	0.096	2.140	0.034	6	y
NUM1	3.666	4.381	0.840	0.404	0	t
NUM2	1310.400	561.354	2.330	0.021	0	d1
NUM3	1158.400	823.990	1.410	0.162	0	d2
NUM4	-10.780	5.515	-1.950	0.053	0	td1
NUM5	-11.383	8.601	-1.320	0.188	0	td2
NUM6	418.976	555.275	0.750	0.452	0	M1
NUM7	154.089	555.941	0.280	0.782	0	M2
NUM8	158.903	556.336	0.290	0.776	0	M3
NUM9	65.601	555.691	0.120	0.906	0	M4
NUM10	9.152	555.545	0.020	0.987	0	M5
NUM11	15.590	556.321	0.030	0.978	0	M6
NUM12	93.935	556.307	0.170	0.866	0	M7
NUM13	173.226	557.313	0.310	0.757	0	M8
NUM14	166.387	558.584	0.300	0.766	0	M9
NUM15	160.655	559.000	0.290	0.774	0	M10
NUM16	131.220	559.237	0.230	0.815	0	M11
NUM17	120.494	559.423	0.220	0.830	0	M12
NUM18	614.979	86.883	7.080	<.0001	0	L1
NUM19	263.361	65.882	4.000	0.000	0	L2
NUM20	-31.323	67.815	-0.460	0.645	0	L3
NUM21	-247.034	49.345	-5.010	<.0001	0	L4
NUM22	270.854	74.203	3.650	0.000	0	L1a
NUM23	385.191	59.600	6.460	<.0001	0	L2a
NUM24	745.943	73.867	10.100	<.0001	0	L3a
NUM25	757.862	63.194	11.990	<.0001	0	L4a

Lampiran 16. Estimasi Parameter ARIMAX *Inflow* Rp.10.000,-

Parameter	Estimasi	SE	T	P	Lag	Variabel
MA1,1	0.209	0.090	2.330	0.021	1	y
MA1,2	-0.341	0.091	-3.730	0.000	11	y
MA1,3	-0.605	0.078	-7.760	<.0001	12	y
MA1,4	-0.652	0.096	-6.810	<.0001	23	y
AR1,1	0.574	0.100	5.730	<.0001	1	y
NUM1	3.601	2.401	1.500	0.136	0	t
NUM2	394.957	308.507	1.280	0.203	0	d1
NUM3	468.900	409.117	1.150	0.254	0	d2
NUM4	-7.650	3.336	-2.290	0.024	0	td1
NUM5	-5.835	4.275	-1.360	0.175	0	td2
NUM6	333.293	306.503	1.090	0.279	0	M1
NUM7	208.462	304.584	0.680	0.495	0	M2
NUM8	189.438	303.375	0.620	0.534	0	M3
NUM9	129.702	304.552	0.430	0.671	0	M4
NUM10	60.653	306.726	0.200	0.844	0	M5
NUM11	-16.108	308.144	-0.050	0.958	0	M6
NUM12	27.495	308.319	0.090	0.929	0	M7
NUM13	157.508	308.407	0.510	0.610	0	M8
NUM14	208.331	309.001	0.670	0.501	0	M9
NUM15	219.224	310.726	0.710	0.482	0	M10
NUM16	207.218	312.300	0.660	0.508	0	M11
NUM17	180.380	310.891	0.580	0.563	0	M12
NUM18	250.041	79.119	3.160	0.002	0	L1
NUM19	37.891	58.135	0.650	0.516	0	L2
NUM20	-90.401	55.012	-1.640	0.103	0	L3
NUM21	-148.377	44.956	-3.300	0.001	0	L4
NUM22	139.235	58.577	2.380	0.019	0	L1a
NUM23	172.037	54.690	3.150	0.002	0	L2a
NUM24	395.256	67.732	5.840	<.0001	0	L3a
NUM25	358.910	58.230	6.160	<.0001	0	L4a

Lampiran 17. Estimasi Parameter ARIMAX *Inflow* Rp.5.000,-

Parameter	Estimasi	SE	T	P	Lag	Variabel
AR1,1	0.371	0.089	4.180	<.0001	1	y
AR1,2	0.510	0.073	7.020	<.0001	11	y
AR1,3	0.406	0.093	4.380	<.0001	12	y
AR1,4	-0.334	0.084	-3.960	0.000	13	y
NUM1	3.981	1.379	2.890	0.005	0	t
NUM2	409.613	181.981	2.250	0.026	0	d1
NUM3	188.322	211.894	0.890	0.376	0	d2
NUM4	-5.172	2.229	-2.320	0.022	0	td1
NUM5	-2.286	2.177	-1.050	0.296	0	td2
NUM6	26.302	181.729	0.140	0.885	0	M1
NUM7	-46.353	182.517	-0.250	0.800	0	M2
NUM8	-68.669	183.062	-0.380	0.708	0	M3
NUM9	-113.077	183.284	-0.620	0.538	0	M4
NUM10	-134.073	183.378	-0.730	0.466	0	M5
NUM11	-136.805	183.628	-0.750	0.458	0	M6
NUM12	-120.901	184.232	-0.660	0.513	0	M7
NUM13	-50.724	185.088	-0.270	0.785	0	M8
NUM14	-26.971	184.444	-0.150	0.884	0	M9
NUM15	-67.067	183.597	-0.370	0.716	0	M10
NUM16	-87.740	183.232	-0.480	0.633	0	M11
NUM17	-102.748	182.687	-0.560	0.575	0	M12
NUM18	149.149	58.665	2.540	0.012	0	L1
NUM19	-28.925	47.108	-0.610	0.540	0	L2
NUM20	-144.784	45.637	-3.170	0.002	0	L3
NUM21	-124.275	35.865	-3.470	0.001	0	L4
NUM22	235.874	45.310	5.210	<.0001	0	L1a
NUM23	128.392	41.041	3.130	0.002	0	L2a
NUM24	123.352	52.177	2.360	0.020	0	L3a
NUM25	111.680	47.238	2.360	0.020	0	L4a

Lampiran 18. Estimasi Parameter ARIMAX *Inflow* Rp.2.000,-

Parameter	Estimasi	SE	T	P	Lag	Variabel
AR1,1	0.444	0.124	3.570	0.001	1	y
AR1,2	-0.442	0.143	-3.080	0.003	14	y
NUM1	1.544	0.345	4.470	<.0001	0	t
NUM2	0.000	0.000	.	.	0	d1
NUM3	-207.687	32.302	-6.430	<.0001	0	d2
NUM4	0.000	0.000	.	.	0	td1
NUM5	9.305	2.528	3.680	0.001	0	td2
NUM6	174.681	21.370	8.170	<.0001	0	M1
NUM7	157.400	21.299	7.390	<.0001	0	M2
NUM8	145.942	21.557	6.770	<.0001	0	M3
NUM9	116.355	21.871	5.320	<.0001	0	M4
NUM10	99.712	22.050	4.520	<.0001	0	M5
NUM11	95.585	22.136	4.320	<.0001	0	M6
NUM12	101.211	22.726	4.450	<.0001	0	M7
NUM13	201.900	24.282	8.310	<.0001	0	M8
NUM14	296.711	25.150	11.800	<.0001	0	M9
NUM15	275.304	23.786	11.570	<.0001	0	M10
NUM16	200.847	23.136	8.680	<.0001	0	M11
NUM17	174.252	23.091	7.550	<.0001	0	M12
NUM18	0.000	0.000	.	.	0	L1
NUM19	-130.696	24.876	-5.250	<.0001	0	L2
NUM20	-115.640	20.690	-5.590	<.0001	0	L3
NUM21	-99.572	24.278	-4.100	0.000	0	L4
NUM22	0.000	0.000	.	.	0	L1a
NUM23	-15.879	24.544	-0.650	0.521	0	L2a
NUM24	-59.856	20.455	-2.930	0.005	0	L3a
NUM25	-101.290	24.416	-4.150	0.000	0	L4a

Lampiran 19. Estimasi Parameter ARIMAX *Inflow* Rp.1.000,-

Parameter	Estimasi	SE	T	P	Variabel
MA1,1	0.179	0.083	2.150	0.034	4
MA1,2	-0.469	0.089	-5.280	<.0001	12
AR1,1	0.187	0.081	2.300	0.023	1
AR1,2	-0.286	0.085	-3.380	0.001	7
AR1,3	0.309	0.086	3.600	0.001	11
NUM1	-0.929	0.184	-5.050	<.0001	0
NUM2	9.860	23.948	0.410	0.681	0
NUM3	36.770	31.319	1.170	0.243	0
NUM4	1.041	0.277	3.760	0.000	0
NUM5	-0.216	0.320	-0.670	0.502	0
NUM6	204.318	25.167	8.120	<.0001	0
NUM7	181.329	25.163	7.210	<.0001	0
NUM8	176.099	25.216	6.980	<.0001	0
NUM9	153.764	25.196	6.100	<.0001	0
NUM10	142.526	25.216	5.650	<.0001	0
NUM11	130.857	25.173	5.200	<.0001	0
NUM12	134.730	25.184	5.350	<.0001	0
NUM13	134.996	25.485	5.300	<.0001	0
NUM14	132.156	25.526	5.180	<.0001	0
NUM15	148.072	25.676	5.770	<.0001	0
NUM16	170.133	26.033	6.540	<.0001	0
NUM17	182.978	25.924	7.060	<.0001	0
NUM18	-7.801	14.152	-0.550	0.583	0
NUM19	-21.502	11.237	-1.910	0.058	0
NUM20	-37.172	12.159	-3.060	0.003	0
NUM21	-57.525	9.506	-6.050	<.0001	0
NUM22	41.069	12.627	3.250	0.002	0
NUM23	34.387	10.432	3.300	0.001	0
NUM24	6.571	12.401	0.530	0.597	0
NUM25	-8.821	10.828	-0.810	0.417	0

Lampiran 20. P value Uji Linieritas (*White Test*) Input DNN-1

P value kurang dari 5% menyimpulkan bahwa hyimpulkan bahwa terdapat hubungan non-linier antara $Y_{i,t}$ dengan lag-nya

Lag $Y_{1,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
4	0.000	0.005	0.000	0.000	0.016	0.000	0.000	0.000	0.003	0.000	100%
5	0.012	0.000	0.014	0.045	0.410	0.015	0.041	0.000	0.330	0.020	80%
7	0.520	0.039	0.210	0.001	0.001	0.380	0.004	0.001	0.350	0.860	50%
12	0.650	0.470	0.044	0.074	0.000	0.320	0.650	0.001	0.900	0.005	40%
13	0.043	0.620	0.067	0.760	0.880	0.000	0.000	0.030	0.650	0.032	50%
14	0.460	0.930	0.066	0.300	0.068	0.063	0.490	0.560	0.930	0.270	0%
15	0.150	0.700	0.110	0.800	0.260	0.480	0.480	0.390	0.000	0.001	20%
17	0.370	0.800	0.001	0.960	0.210	0.270	0.037	0.029	0.003	0.001	50%
23	0.680	0.200	0.180	0.068	0.130	0.400	0.000	0.051	0.260	0.200	10%
24	0.001	0.430	0.670	0.001	0.110	0.370	0.009	0.007	0.001	0.380	50%

Lag $Y_{2,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
3	0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.668	0.230	80%
12	0.160	0.280	0.442	0.680	0.320	0.950	0.319	0.720	0.498	0.027	10%
14	0.940	0.180	0.223	0.780	0.380	0.590	0.368	0.001	0.420	0.888	10%
15	0.720	0.190	0.475	0.820	0.790	0.600	0.911	0.590	0.018	0.250	10%
23	0.660	0.100	0.325	0.650	0.930	0.002	0.502	0.270	0.705	0.002	20%

Lag $Y_{3,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
11	0.750	0.300	0.088	0.320	0.700	0.280	0.250	0.019	0.100	0.710	10%
12	0.000	0.150	0.010	0.520	0.600	0.130	0.330	0.093	0.700	0.340	20%
13	0.770	0.190	0.940	0.210	0.970	0.026	0.930	0.300	0.018	0.023	30%
23	0.640	0.940	0.130	0.004	0.600	0.330	0.033	0.140	0.000	0.650	30%

Lag $Y_{4,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
11	0.010	0.510	0.006	0.003	0.100	0.100	0.880	0.190	0.082	0.002	40%
12	0.100	0.001	0.540	0.046	0.860	0.120	0.260	0.030	0.160	0.790	30%
13	0.800	0.069	0.690	0.310	0.093	0.000	0.850	0.920	0.003	0.540	20%
22	0.820	0.400	0.008	0.310	0.330	0.110	0.200	0.025	0.150	0.690	20%
23	0.280	0.300	0.340	0.150	0.530	0.330	0.002	0.590	0.340	0.010	20%

Lag $Y_{5,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
8	0.073	0.002	0.057	0.004	0.450	0.170	0.012	0.001	0.020	0.004	60%
11	0.160	0.440	0.400	0.054	0.000	0.050	0.650	0.180	0.170	0.025	20%
12	0.004	0.740	0.061	0.710	0.001	0.017	0.005	0.015	0.000	0.110	60%
22	0.370	0.710	0.350	0.012	0.200	0.008	0.096	0.970	0.250	0.027	30%
23	0.230	0.560	0.034	0.340	0.018	0.690	0.059	0.300	0.800	0.460	20%
24	0.810	0.310	0.079	0.003	0.320	0.800	0.270	0.160	0.550	0.420	10%
Lag $Y_{6,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
8	0.007	0.239	0.001	0.064	0.000	0.000	0.000	0.434	0.000	0.013	70%
9	0.002	0.148	0.001	0.018	0.210	0.170	0.018	0.007	0.019	0.100	60%
10	0.016	0.247	0.000	0.010	0.003	0.009	0.000	0.013	0.001	0.002	90%
12	0.686	0.993	0.025	0.813	0.220	0.230	0.330	0.202	0.001	0.005	30%
22	0.002	0.009	0.000	0.000	0.023	0.180	0.005	0.028	0.000	0.000	90%
Lag $Y_{7,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	100%
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	100%
11	0.002	0.109	0.018	0.220	0.000	0.000	0.350	0.000	0.180	0.500	50%
12	0.099	0.612	0.010	0.051	0.003	0.220	0.000	0.240	0.072	0.130	30%
13	0.150	0.871	0.035	0.300	0.220	0.310	0.470	0.004	0.022	0.018	40%
14	0.190	0.039	0.730	0.960	0.790	0.120	0.003	0.650	0.580	0.003	30%
23	0.550	0.240	0.100	0.130	0.420	0.004	0.730	0.007	0.210	0.540	20%
24	0.420	0.305	0.700	0.390	0.120	0.890	0.310	0.980	0.980	0.070	0%
Lag $Y_{8,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
3	0.000	0.130	0.160	0.510	0.000	0.000	0.520	0.007	0.045	0.000	60%
4	0.260	0.000	0.000	0.002	0.320	0.001	0.000	0.000	0.000	0.000	80%
5	0.093	0.730	0.880	0.007	0.000	0.072	0.057	0.014	0.190	0.005	40%
6	0.190	0.920	0.005	0.019	0.510	0.860	0.083	0.530	0.420	0.840	20%
12	0.024	0.720	0.067	0.099	0.170	0.530	0.180	0.550	0.091	0.210	10%
14	0.560	0.000	0.450	0.036	0.250	0.017	0.320	0.000	0.350	0.670	40%
23	0.001	0.130	0.470	0.016	0.420	0.430	0.350	0.860	0.220	0.250	20%
Lag $Y_{9,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.060	0.060	0.067	0.064	0.061	0.064	0.060	0.060	0.062	0.061	0%
2	0.002	0.004	0.007	0.002	0.004	0.016	0.141	0.028	0.003	0.079	80%
3	0.012	0.000	0.008	0.346	0.029	0.680	0.067	0.675	0.000	0.078	50%
12	0.930	0.800	0.712	0.496	0.352	0.008	0.318	0.003	0.731	0.889	20%
13	0.095	0.810	0.919	0.162	0.211	0.000	0.593	0.458	0.322	0.376	10%
14	0.000	0.430	0.636	0.351	0.563	0.880	0.238	0.464	0.748	0.572	10%
15	0.360	0.420	0.568	0.647	0.641	0.700	0.698	0.583	0.529	0.843	0%
23	0.015	0.960	0.086	0.999	0.657	0.700	0.030	0.420	0.073	0.103	20%

<i>Lag</i> $Y_{10,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.033	0.014	0.025	0.025	0.151	0.043	0.012	0.032	0.023	0.054	80%
2	0.711	0.659	0.910	0.030	0.332	0.469	0.813	0.877	0.727	0.876	10%
3	0.032	0.508	0.597	0.090	0.011	0.291	0.000	0.335	0.167	0.451	30%
11	0.748	0.575	0.353	0.786	0.397	0.971	0.039	0.056	0.392	0.835	10%
12	0.927	0.312	0.797	0.760	0.145	0.307	0.637	0.380	0.127	0.295	0%
13	0.704	0.310	0.398	0.004	0.586	0.051	0.011	0.752	0.336	0.466	20%
14	0.064	0.603	0.089	0.438	0.569	0.224	0.209	0.370	0.696	0.157	0%
15	0.967	0.690	0.887	0.049	0.815	0.409	0.716	0.120	0.782	0.112	10%

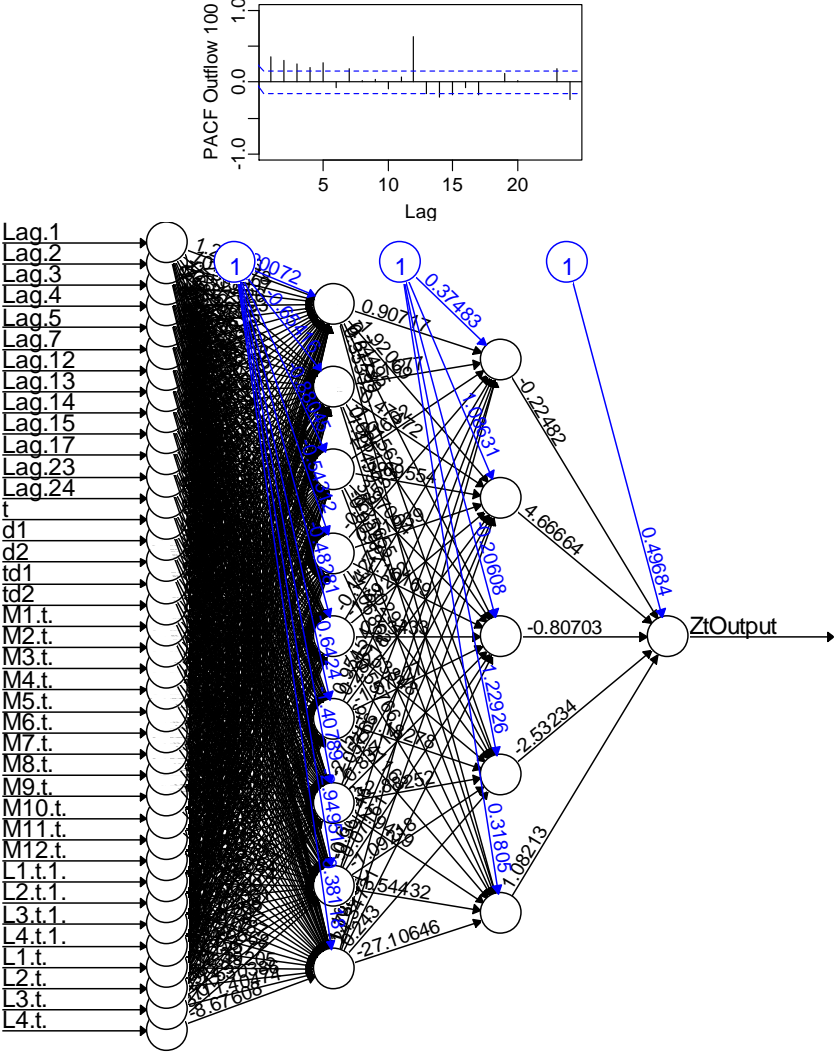
<i>Lag</i> $Y_{11,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.573	0.350	0.814	0.742	0.780	0.540	0.493	0.870	0.929	0.960	0%
11	0.147	0.150	0.039	0.368	0.930	0.750	0.318	0.300	0.070	0.001	20%
12	0.259	0.630	0.075	0.929	0.690	0.770	0.048	0.600	0.061	0.875	10%
13	0.049	0.840	0.432	0.088	0.180	0.620	0.391	0.290	0.007	0.547	20%

<i>Lag</i> $Y_{12,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.047	0.074	0.056	0.049	0.091	0.090	0.098	0.043	0.053	0.041	40%
2	0.120	0.011	0.953	0.711	0.156	0.590	0.249	0.276	0.004	0.526	20%
11	0.756	0.100	0.564	0.350	0.111	0.910	0.573	0.604	0.789	0.622	0%
12	0.057	0.028	0.414	0.116	0.155	0.150	0.451	0.477	0.968	0.298	10%
13	0.841	0.980	0.546	0.455	0.238	0.760	0.868	0.073	0.844	0.950	0%

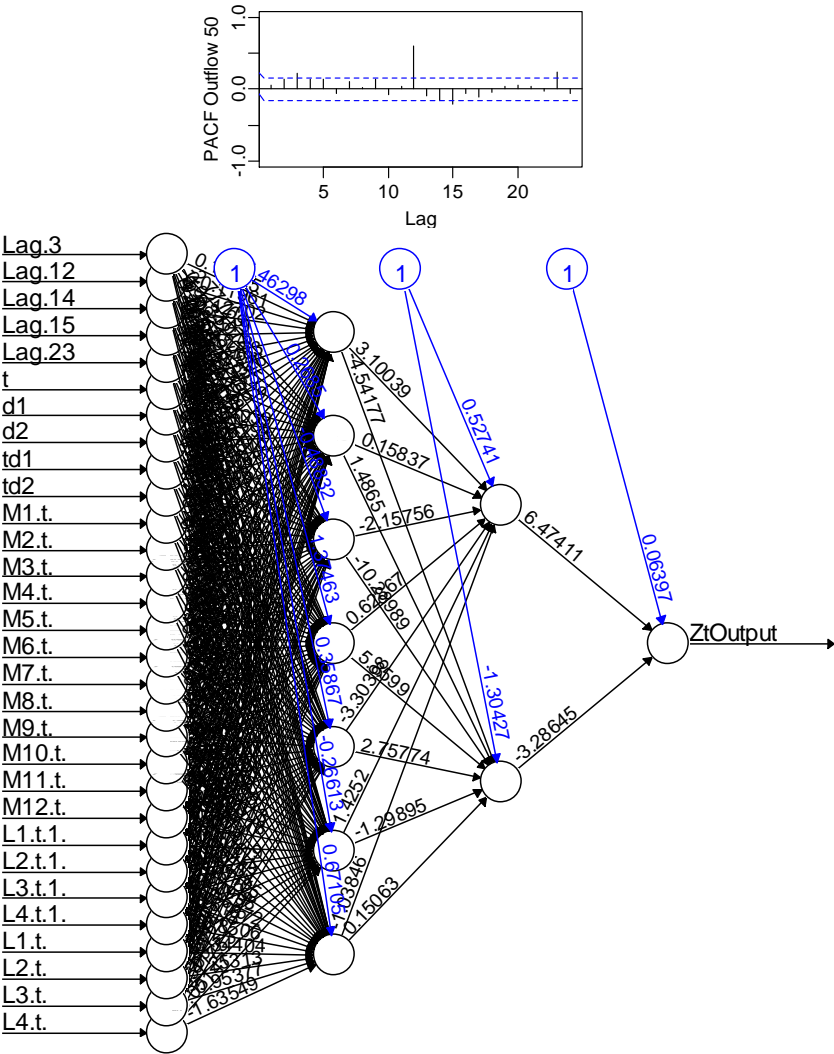
<i>Lag</i> $Y_{13,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.041	0.060	0.089	0.048	0.097	0.072	0.057	0.071	0.059	0.105	20%
2	0.776	0.310	0.169	0.017	0.519	0.028	0.031	0.008	0.130	0.010	50%
11	0.443	0.960	0.335	0.002	0.007	0.023	0.488	0.616	0.390	0.262	30%
12	0.000	0.400	0.001	0.213	0.031	0.088	0.061	0.432	0.000	0.013	50%
13	0.124	0.520	0.025	0.037	0.908	0.047	0.001	0.655	0.330	0.197	40%

<i>Lag</i> $Y_{14,t}$	Ulangan ke-										% (Non-linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.003	0.002	0.003	0.003	0.003	0.003	0.002	0.002	0.004	0.002	100%
2	0.042	0.000	0.056	0.013	0.001	0.001	0.005	0.000	0.003	0.000	90%
8	0.429	0.250	0.386	0.029	0.520	0.014	0.867	0.096	0.083	0.005	30%
9	0.744	0.460	0.446	0.749	0.086	0.950	0.712	0.007	0.280	0.470	10%
11	0.607	0.250	0.266	0.525	0.001	0.980	0.187	0.060	0.265	0.058	10%
12	0.247	0.000	0.644	0.005	0.000	0.001	0.001	0.410	0.465	0.950	50%
13	0.705	0.240	0.183	0.213	0.004	0.530	0.006	0.001	0.089	0.000	40%
14	0.010	0.230	0.412	0.228	0.280	0.000	0.428	0.001	0.571	0.170	30%

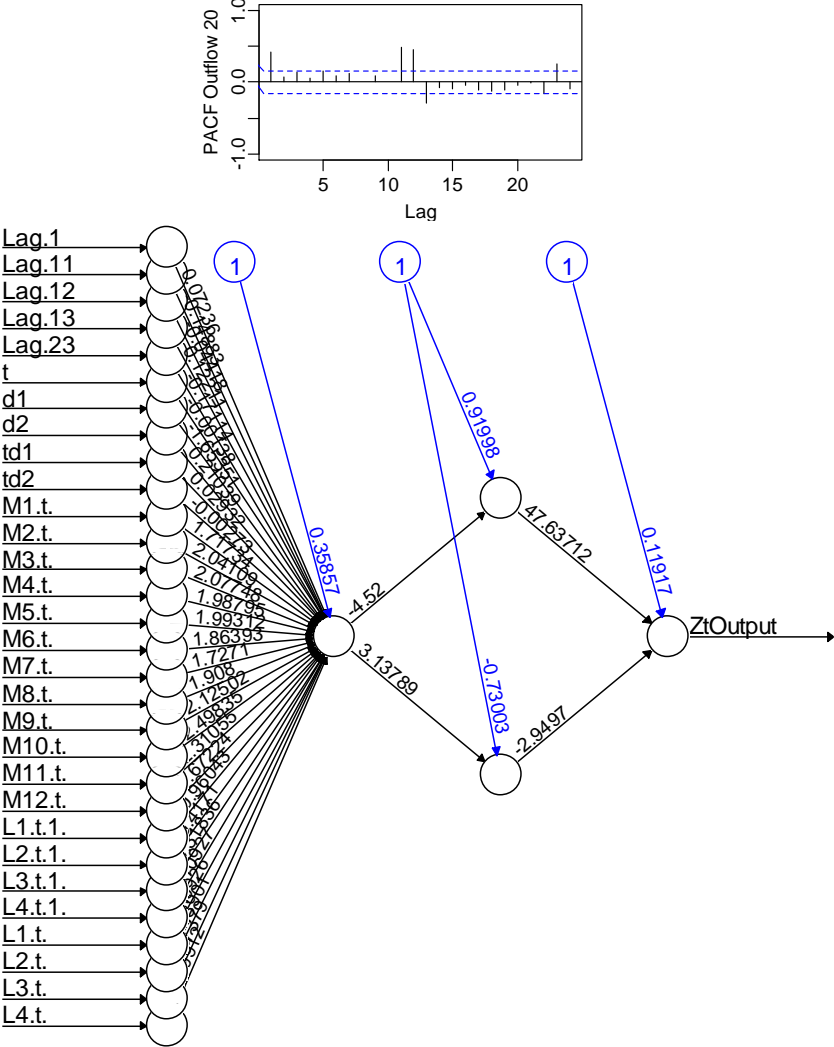
Lampiran 21. DNN-1 Outflow Rp 100.000,-



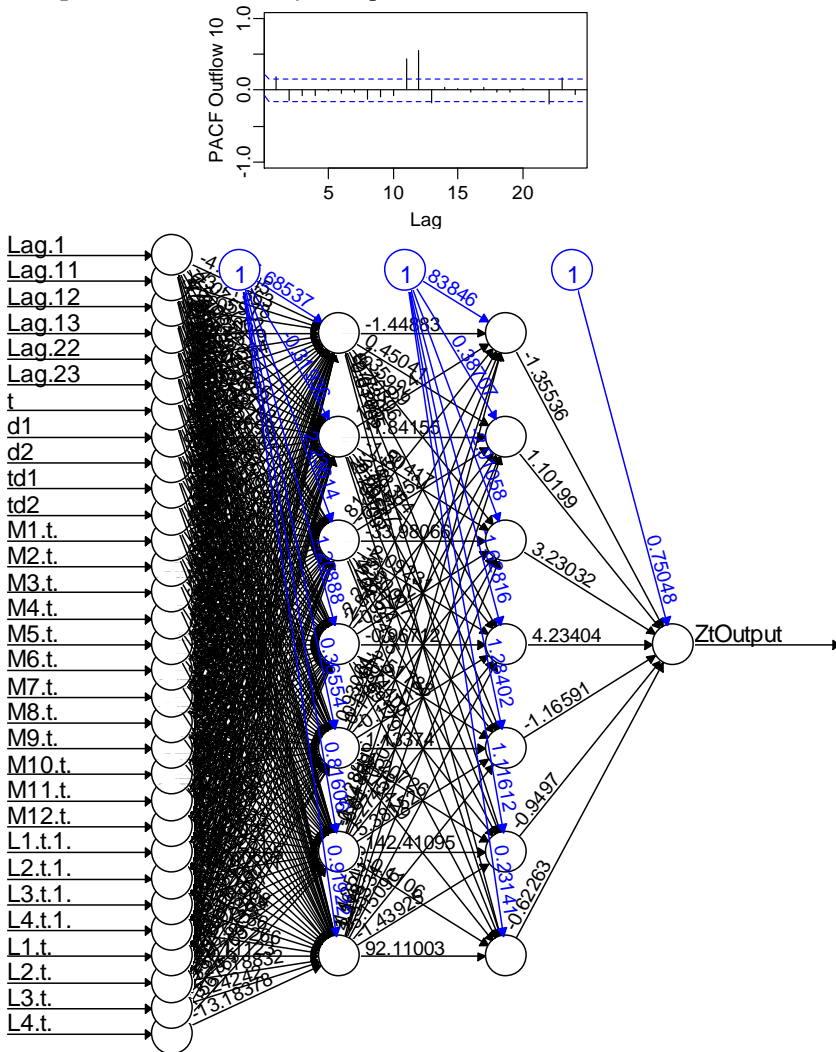
Lampiran 22. DNN-1 *Outflow* Rp 50.000,-



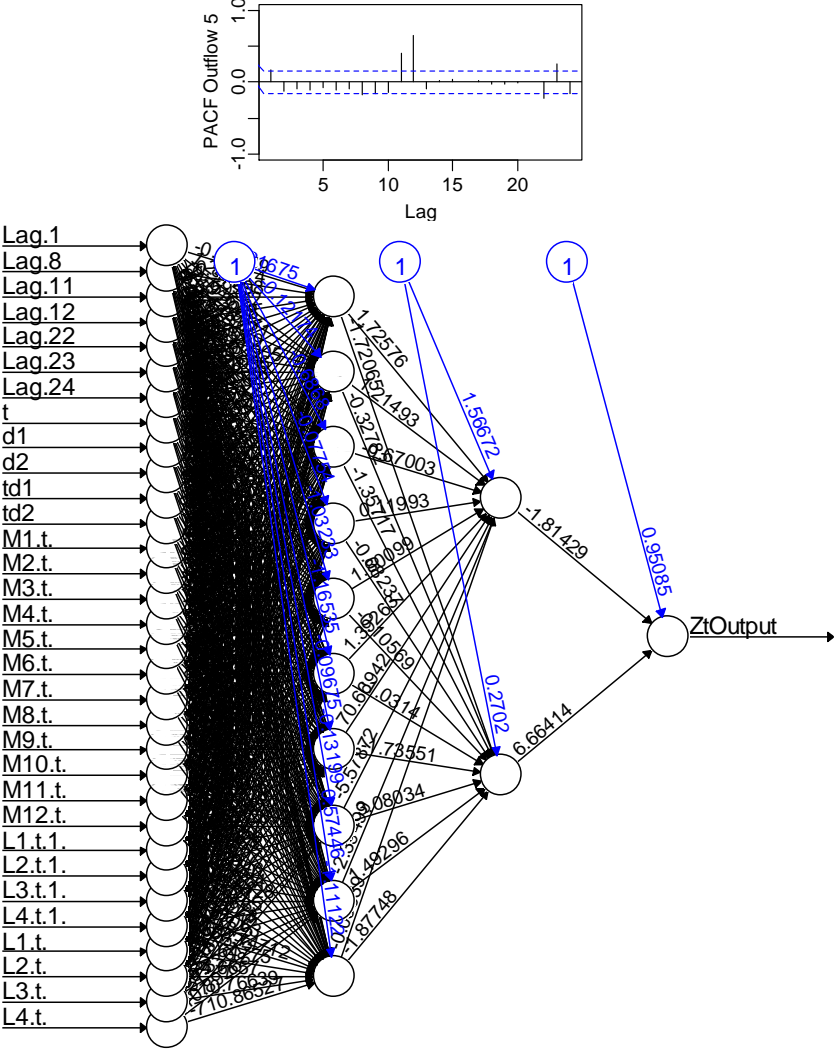
Lampiran 23. DNN-1 Outflow Rp 20.000,-



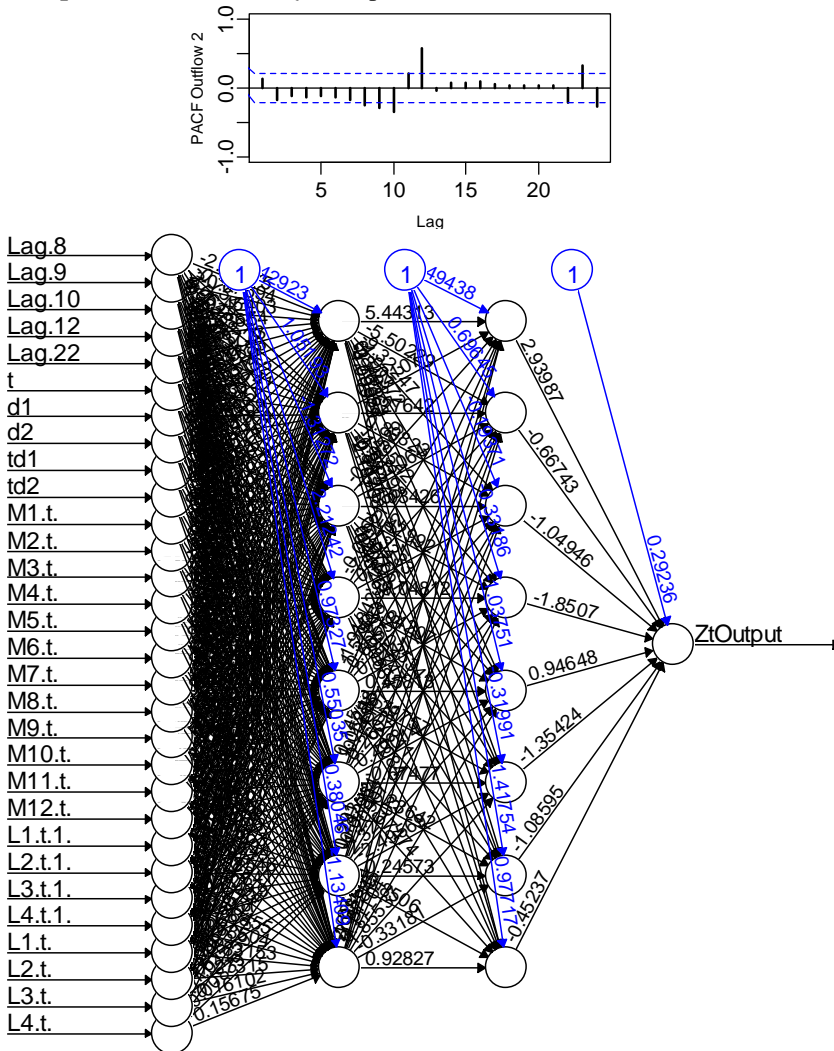
Lampiran 24. DNN-1 Outflow Rp 10.000,-



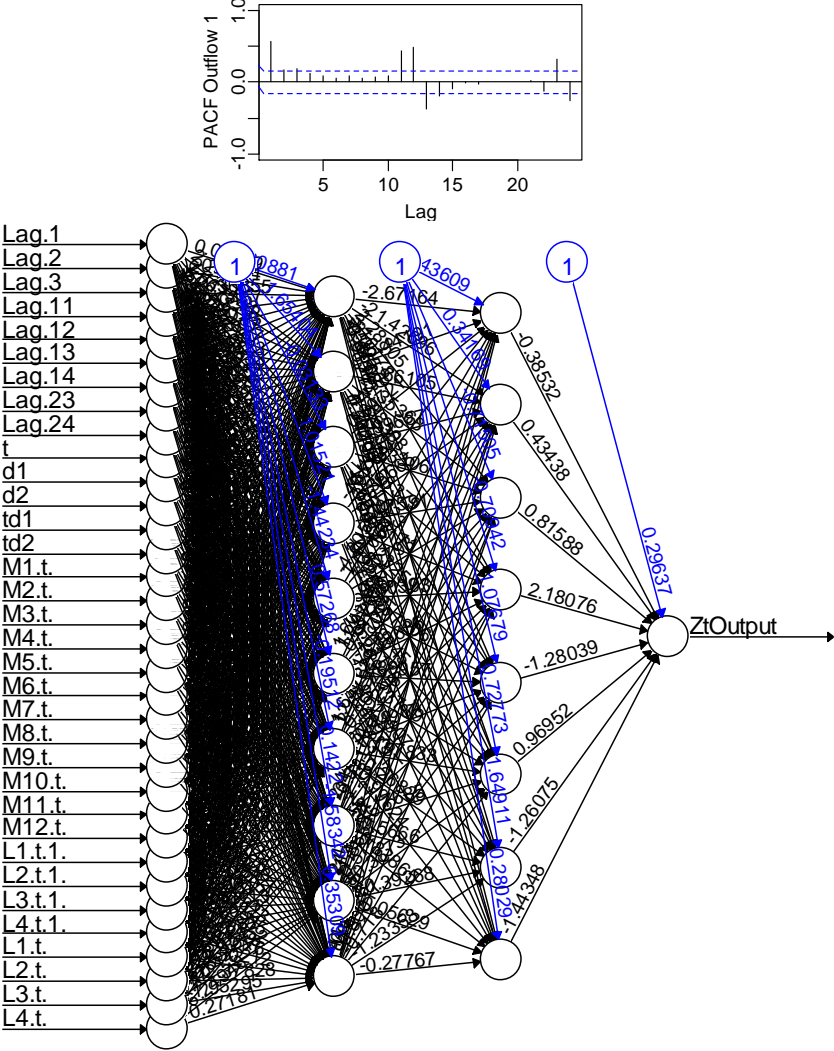
Lampiran 25. DNN-1 Outflow Rp 5.000,-



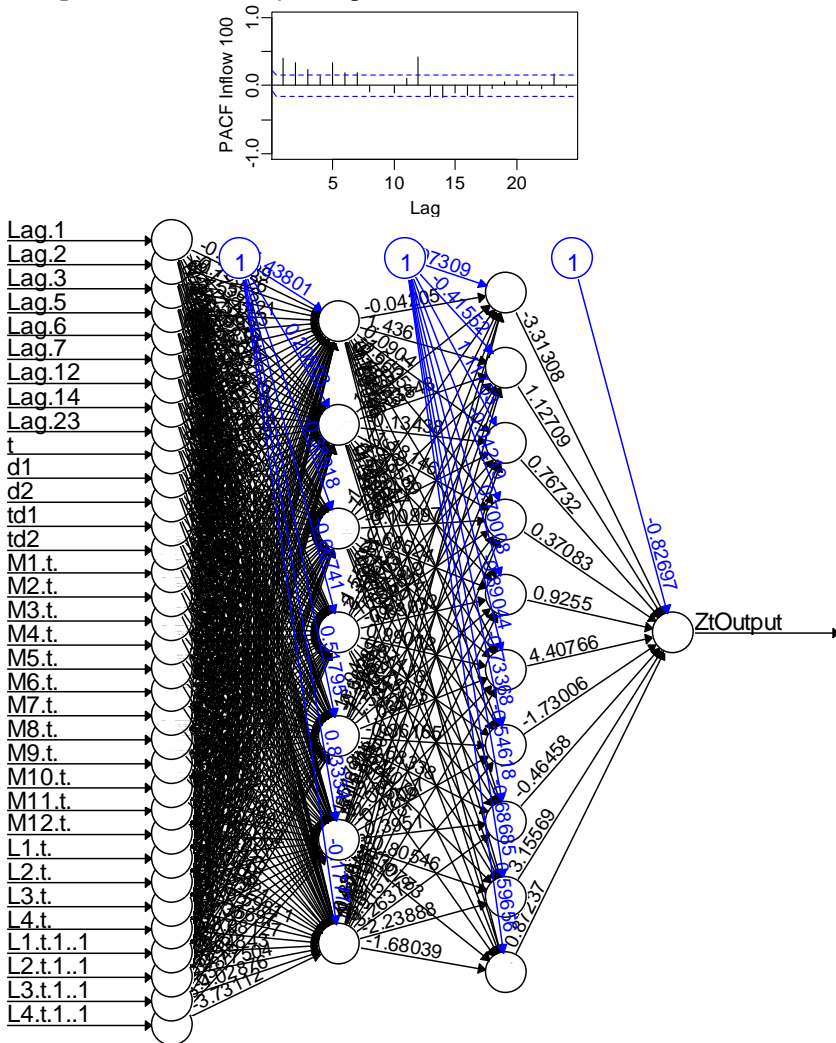
Lampiran 26. DNN-1 Outflow Rp.2.000,-



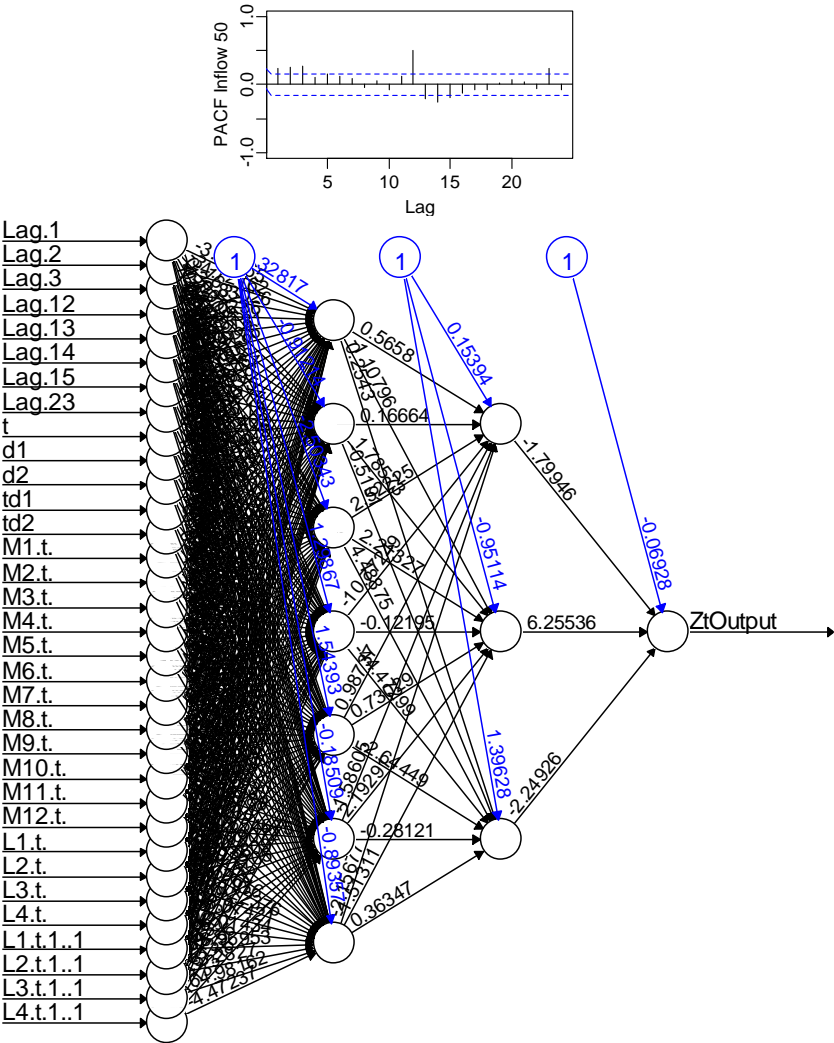
Lampiran 27. DNN-1 Outflow Rp 1.000,-



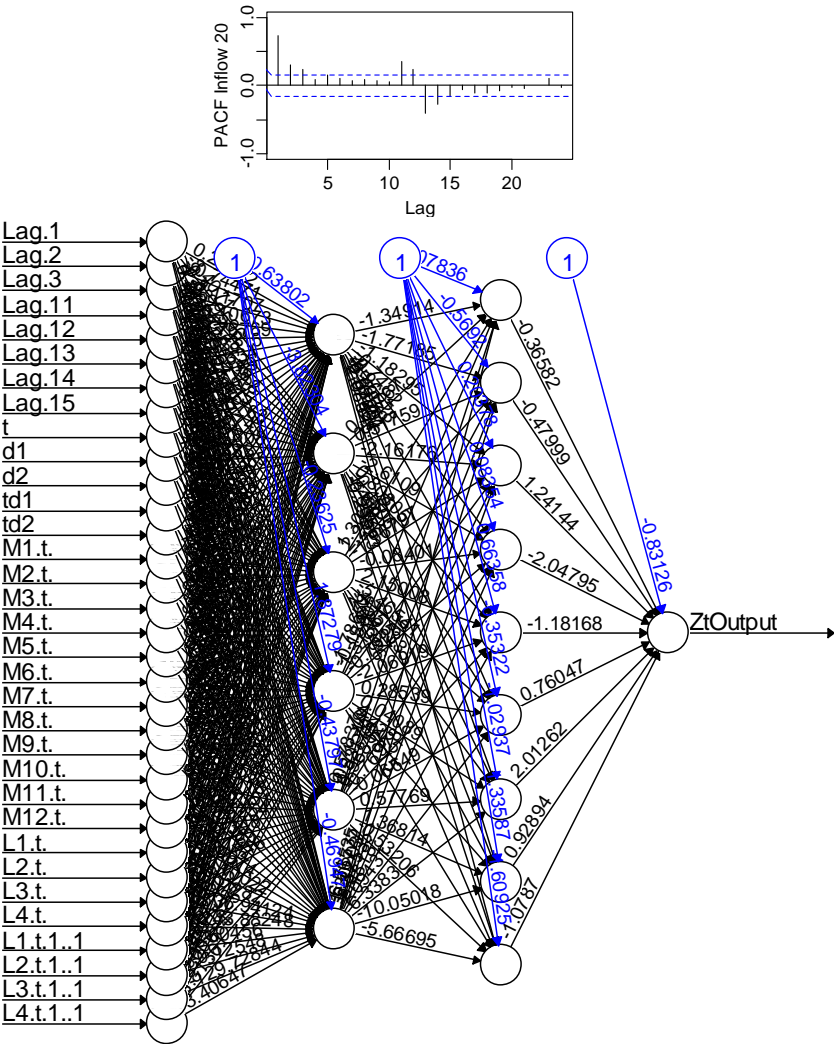
Lampiran 28. DNN-1 Inflow Rp 100.000,-



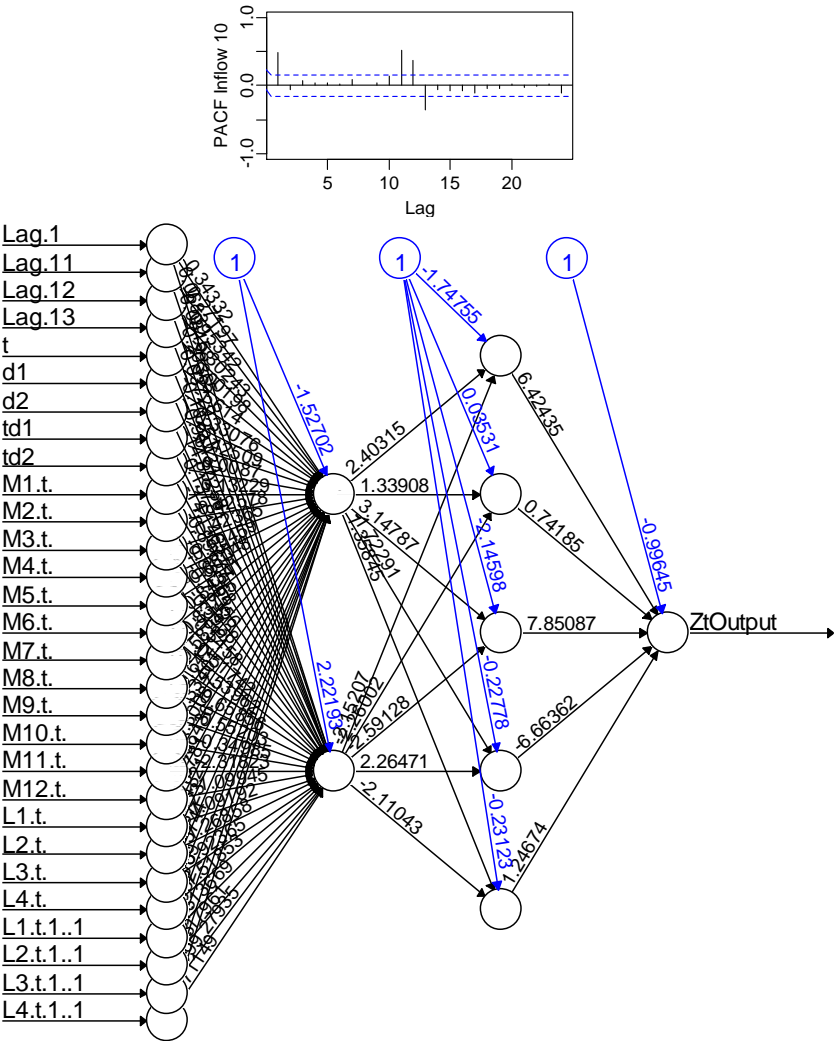
Lampiran 29. DNN-1 *Inflow* Rp 50.000,-



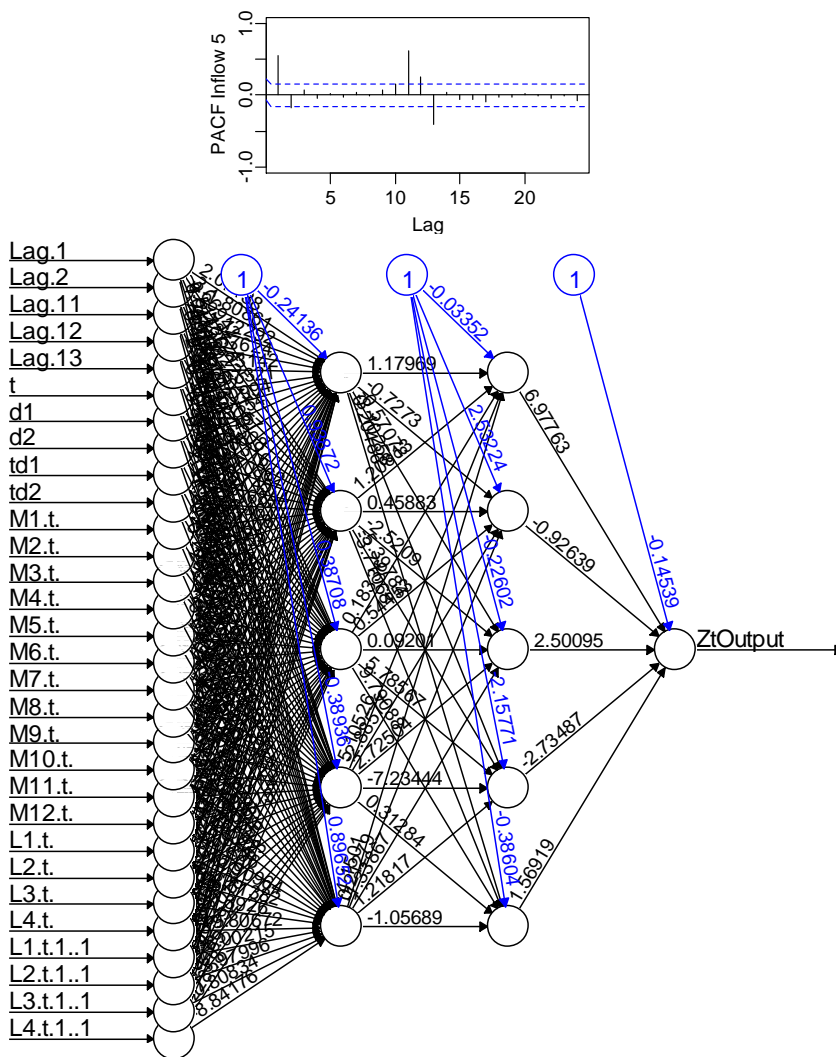
Lampiran 30. DNN-1 Inflow Rp 20.000,-



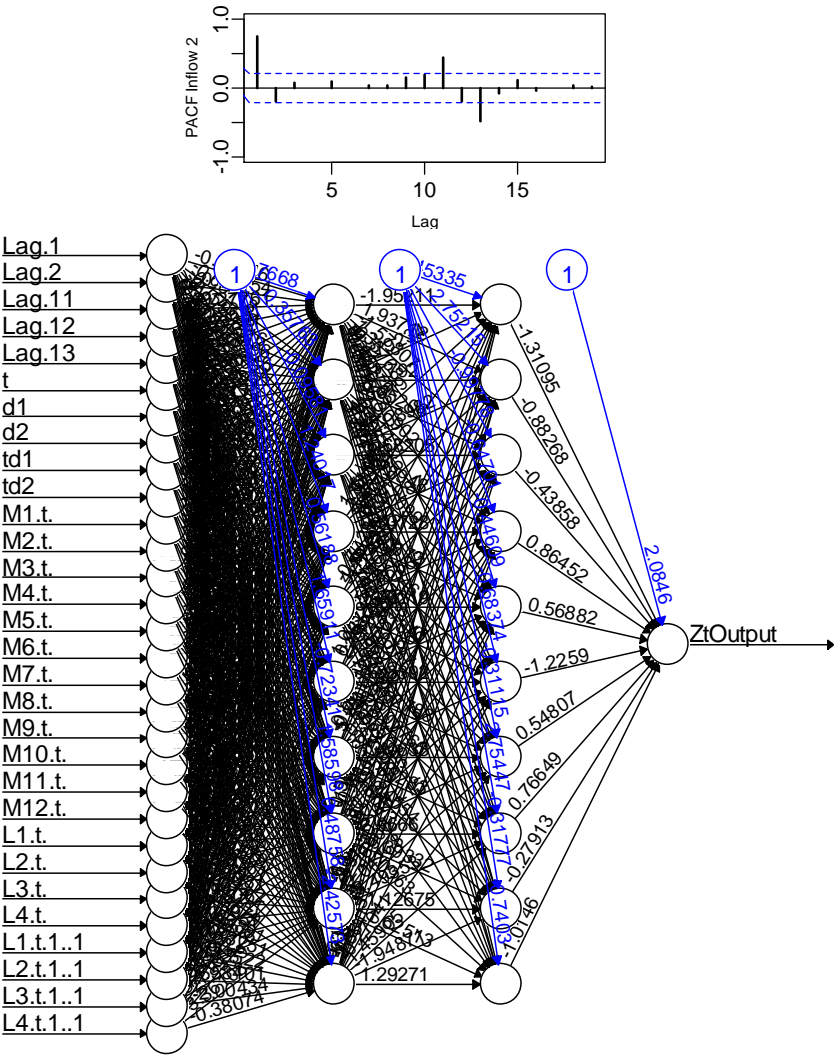
Lampiran 31. DNN-1 Inflow Rp 10.000,-



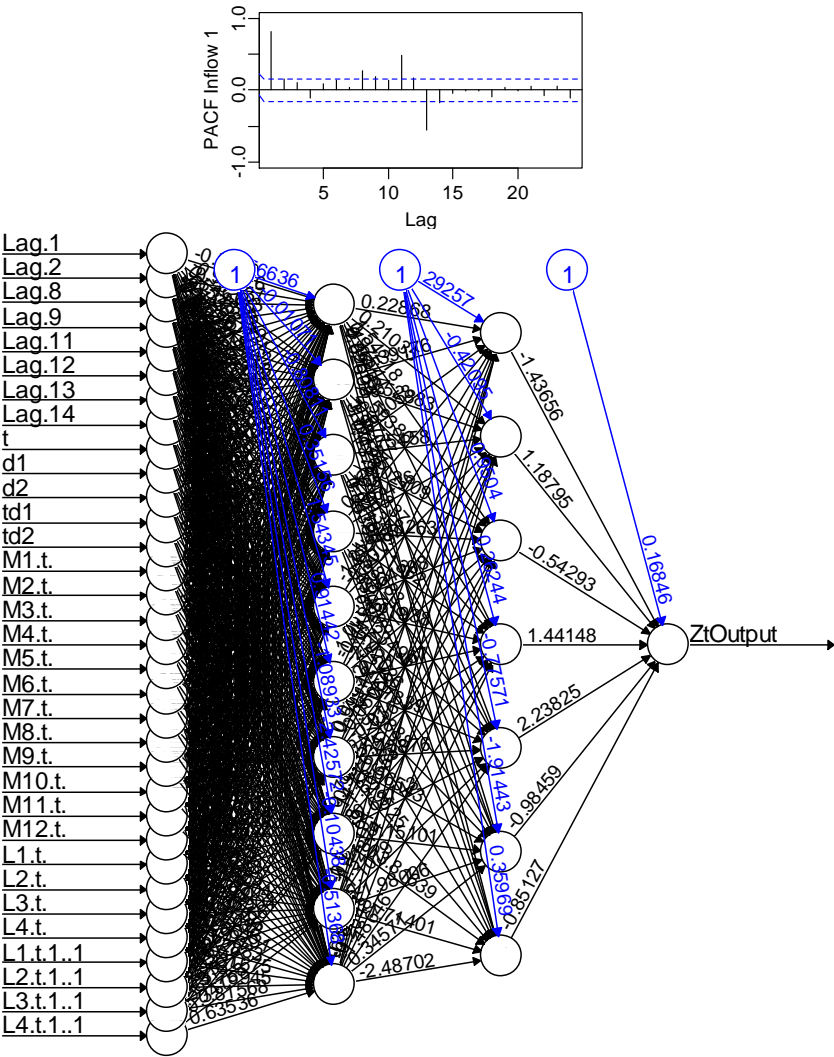
Lampiran 32. DNN-1 Inflow Rp 5.000,-



Lampiran 33. DNN-1 Inflow Rp.2.000,-



Lampiran 34. DNN-1 Inflow Rp 1.000,-



Lampiran 35. P value Uji Linieritas (*White Test*) Input DNN-2

P value kurang dari 5% menyimpulkan bahwa terdapat hubungan non-linier antara $Y_{i,t}$ dengan lag-nya. Adanya hubungan non-linier disimpulkan apabila minimal terdapat 1 lag yang menunjukkan % non-linieritas lebih dari sama dengan 50%

Lag $Y_{1,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
10	0.100	0.099	0.990	0.054	0.210	0.737	0.670	0.187	0.760	0.129	0%	Ya
12	0.170	0.063	0.010	0.031	0.020	0.002	0.110	0.001	0.001	0.056	60%	

Lag $Y_{2,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
12	0.840	0.400	0.160	0.112	0.034	0.019	0.860	0.940	0.012	0.103	30%	Tidak
13	0.620	0.580	0.390	0.440	0.229	0.341	0.540	0.290	0.242	0.871	0%	
23	0.460	0.280	0.270	0.366	0.965	0.125	0.460	0.200	0.547	0.058	0%	
26	0.380	0.820	0.330	0.046	0.893	0.635	0.580	0.670	0.797	0.448	10%	

Lag $Y_{3,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
23	0.200	0.190	0.360	0.920	0.180	0.059	0.580	0.820	0.310	0.810	0%	Tidak

Lag $Y_{4,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
22	0.130	0.210	0.870	0.390	0.030	0.590	0.380	0.850	0.890	0.090	10%	Tidak

Lag $Y_{5,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%	Ya
12	0.077	0.012	0.190	0.660	0.120	0.820	0.002	0.340	0.290	0.380	20%	
23	0.310	0.400	0.045	0.850	0.066	0.110	0.280	0.004	0.800	0.370	20%	

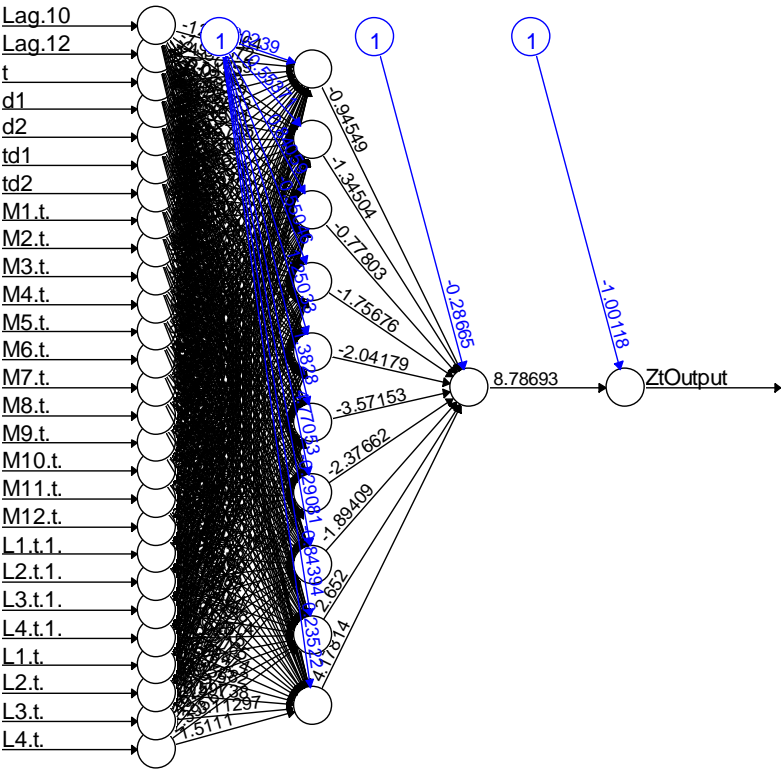
Lag $Y_{6,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%	Ya

Lag $Y_{7,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%	Ya
11	0.022	0.450	0.026	0.031	0.039	0.003	0.111	0.000	0.760	0.010	70%	
13	0.270	0.200	0.262	0.360	0.280	0.890	0.004	0.829	0.070	0.940	10%	

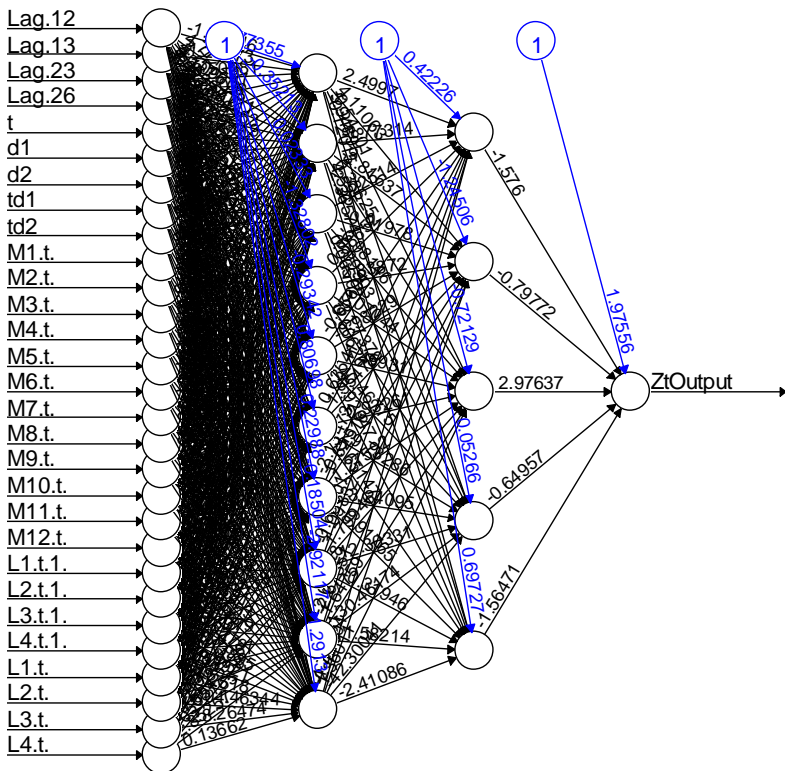
Lag $Y_{8,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%	Ya
12	0.006	0.180	0.026	0.063	0.110	0.210	0.074	0.320	0.600	0.110	20%	

$Lag Y_{9,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
3	0.093	0.120	0.007	0.220	0.001	0.020	0.530	0.170	0.006	0.011	50%	Ya
16	0.998	0.910	0.698	0.392	0.703	0.121	0.040	0.690	0.140	0.385	10%	
18	0.679	0.760	0.400	0.021	0.513	0.128	0.350	0.750	0.685	0.336	10%	
24	0.135	0.200	0.602	0.062	0.114	0.063	0.770	0.700	0.848	0.185	0%	
25	0.919	0.720	0.066	0.589	0.296	0.328	0.540	0.030	0.309	0.045	20%	
$Lag Y_{10,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
1	0.023	0.020	0.080	0.013	0.029	0.054	0.006	0.037	0.021	0.055	70%	Ya
2	0.708	0.900	0.860	0.829	0.429	0.806	0.943	0.122	0.249	0.953	0%	
6	0.977	0.980	0.840	0.577	0.721	0.432	0.454	0.033	0.692	0.434	10%	
$Lag Y_{11,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
1	0.650	0.840	0.720	0.280	0.800	0.730	0.740	0.680	0.650	0.640	0%	Tidak
$Lag Y_{12,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
1	0.040	0.056	0.062	0.037	0.056	0.044	0.036	0.042	0.038	0.074	60%	Ya
11	0.150	0.603	0.133	0.032	0.315	0.885	0.908	0.065	0.723	0.398	10%	
12	0.210	0.781	0.764	0.254	0.900	0.978	0.616	0.197	0.008	0.111	10%	
13	0.460	0.663	0.185	0.608	0.658	0.879	0.326	0.635	0.929	0.470	0%	
$Lag Y_{13,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
1	0.073	0.062	0.061	0.047	0.055	0.018	0.046	0.045	0.073	0.046	50%	Ya
14	0.050	0.037	0.010	0.145	0.267	0.584	0.173	0.711	0.022	0.325	30%	
$Lag Y_{14,t}$	Ulangan ke-										% (Non-linier)	Non Linier
	1	2	3	4	5	6	7	8	9	10		
1	0.004	0.002	0.002	0.003	0.003	0.002	0.004	0.002	0.004	0.003	100%	Ya
7	0.877	0.448	0.492	0.270	0.249	0.397	0.744	0.086	0.391	0.039	10%	
11	0.009	0.572	0.694	0.251	0.153	0.209	0.058	0.496	0.094	0.414	10%	

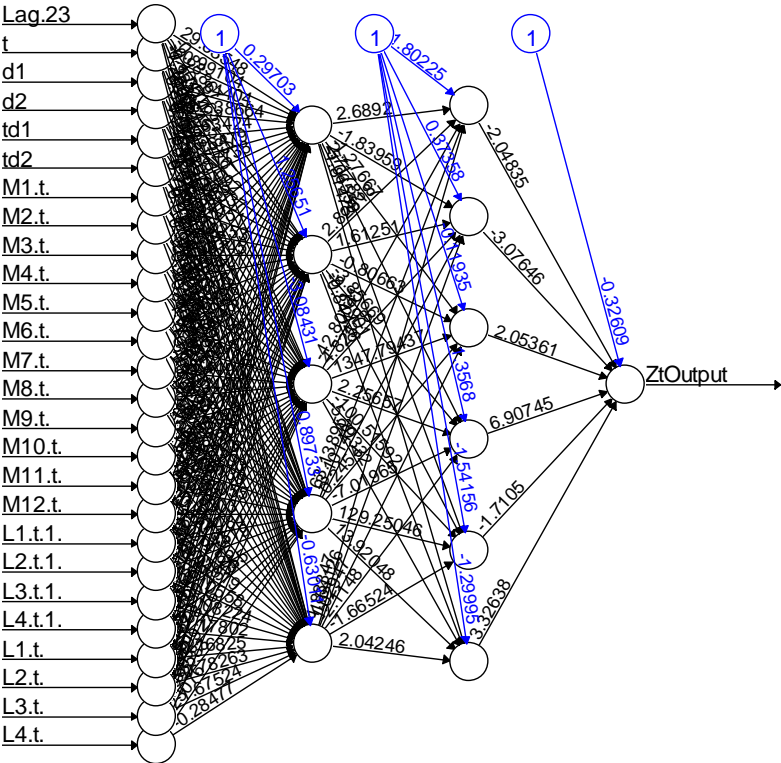
Lampiran 36. DNN-2 Outflow Rp 100.000,-



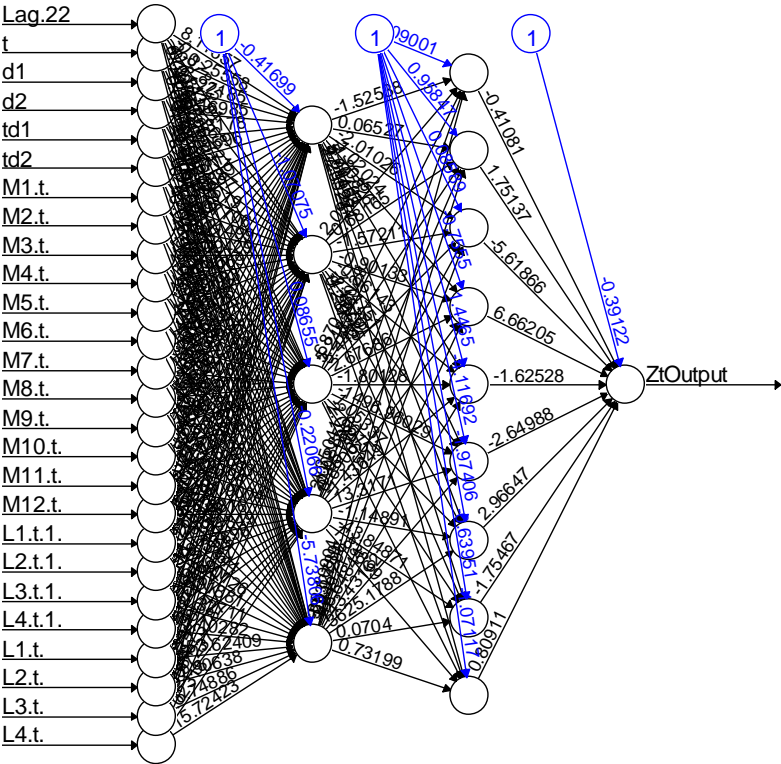
Lampiran 37. DNN-2 Outflow Rp 50.000,-



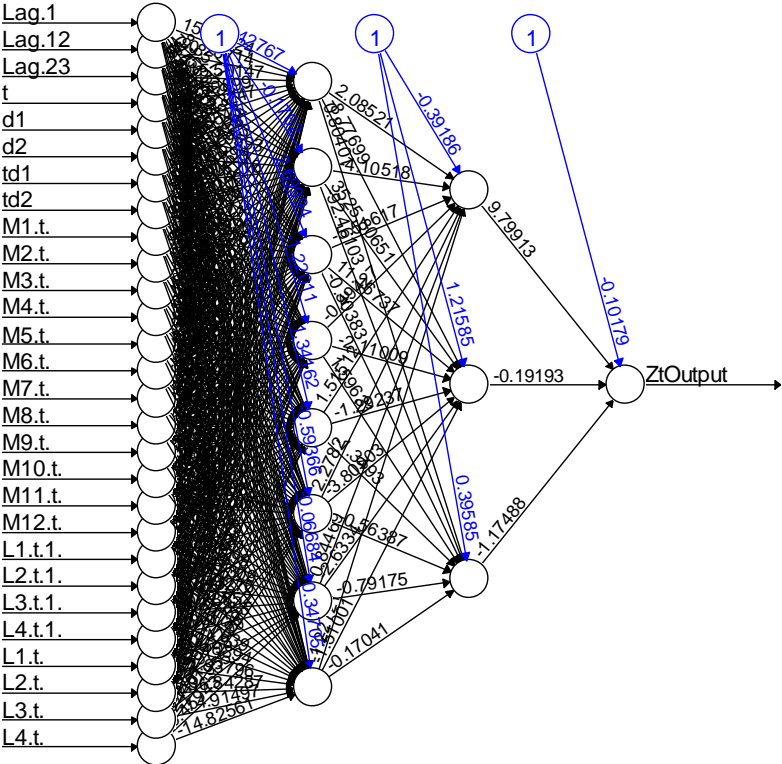
Lampiran 38. DNN-2 Outflow Rp 20.000,-



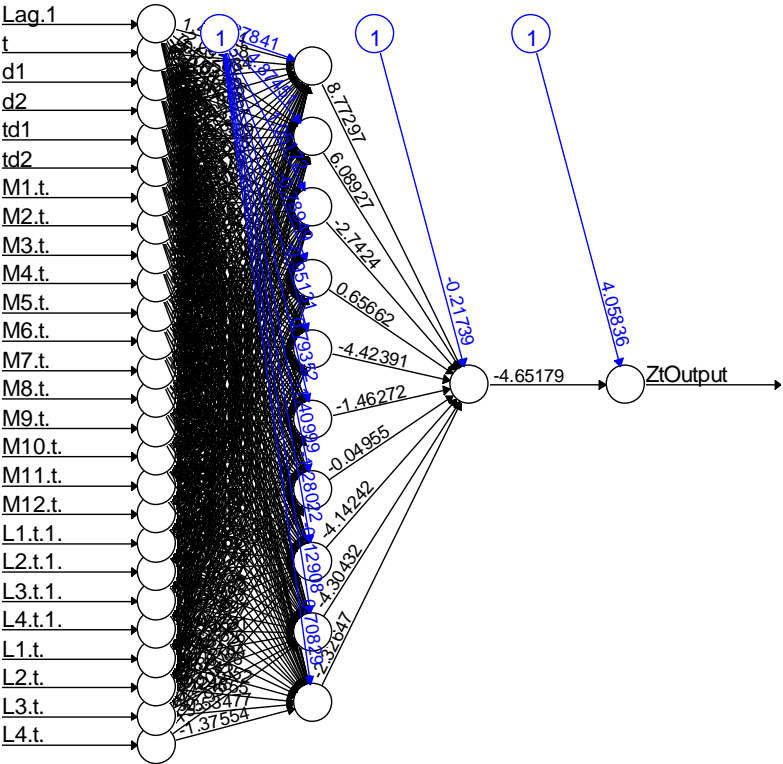
Lampiran 39. DNN-2 Outflow Rp 10.000,-



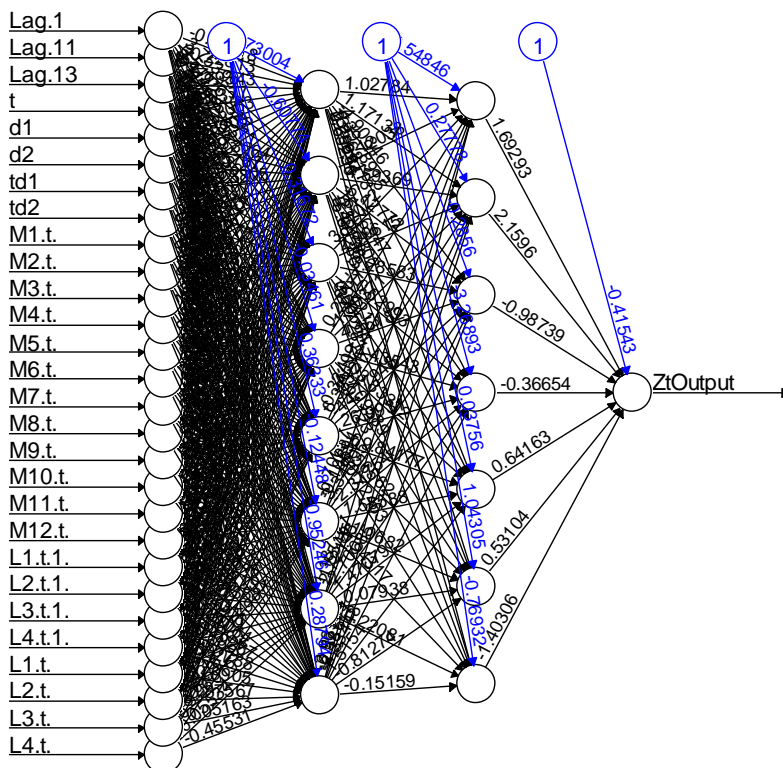
Lampiran 40. DNN-2 Outflow Rp 5.000,-



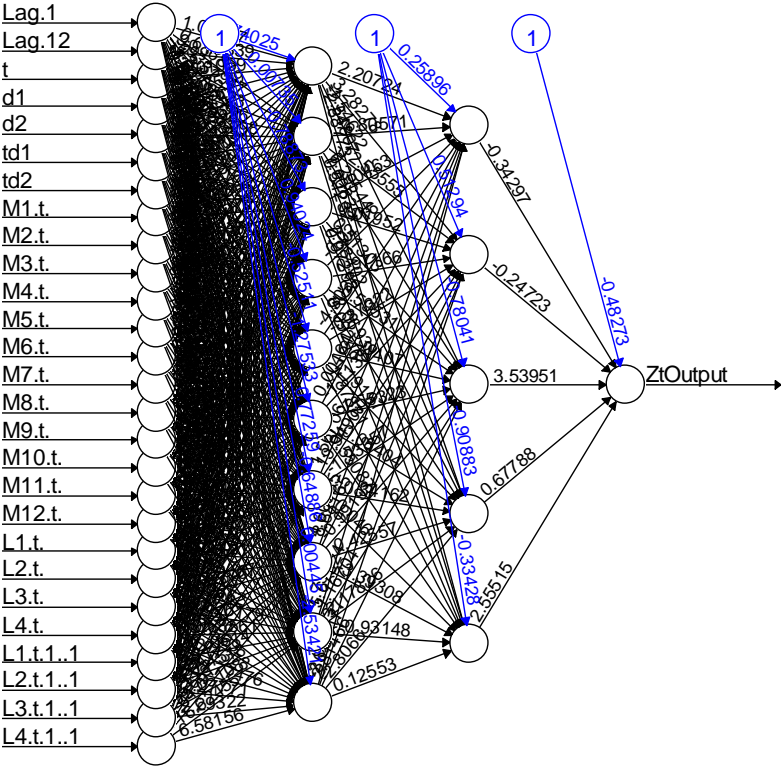
Lampiran 41.DNN-2 Outflow Rp 2.000,-



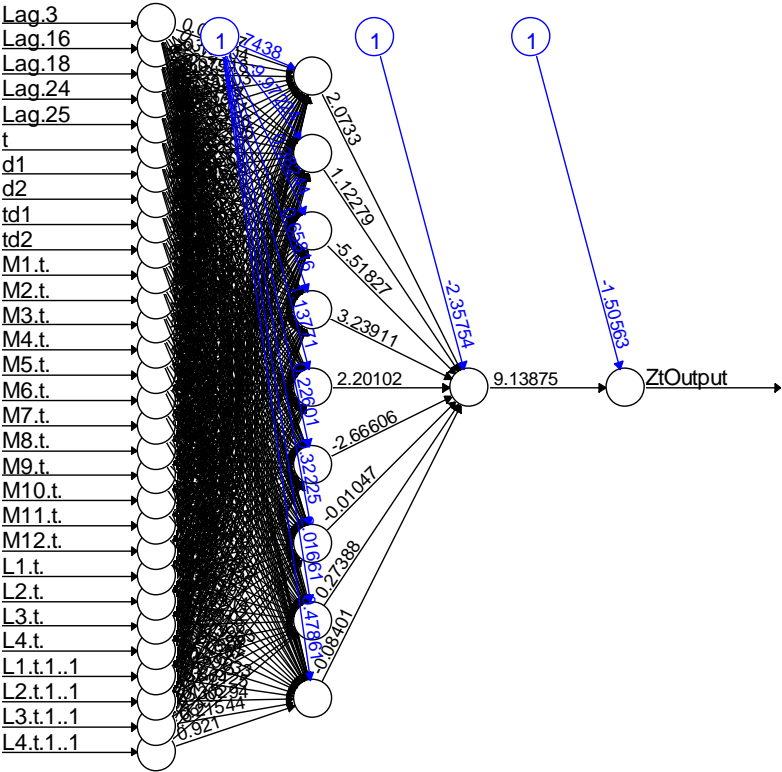
Lampiran 42. DNN-2 Outflow Rp 1.000,-



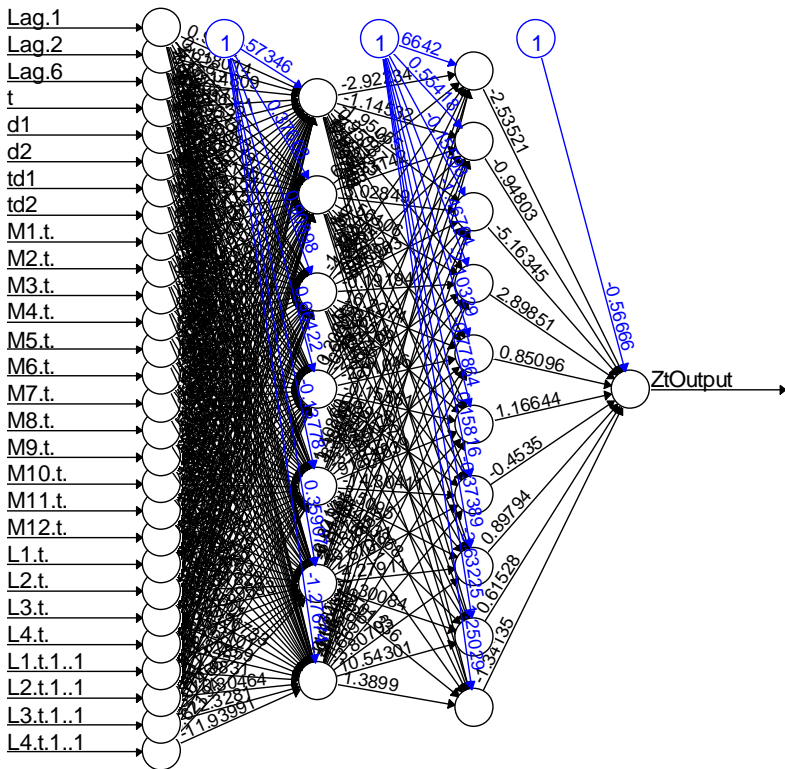
Lampiran 43.DNN-2 Inflow Rp 100.000,-



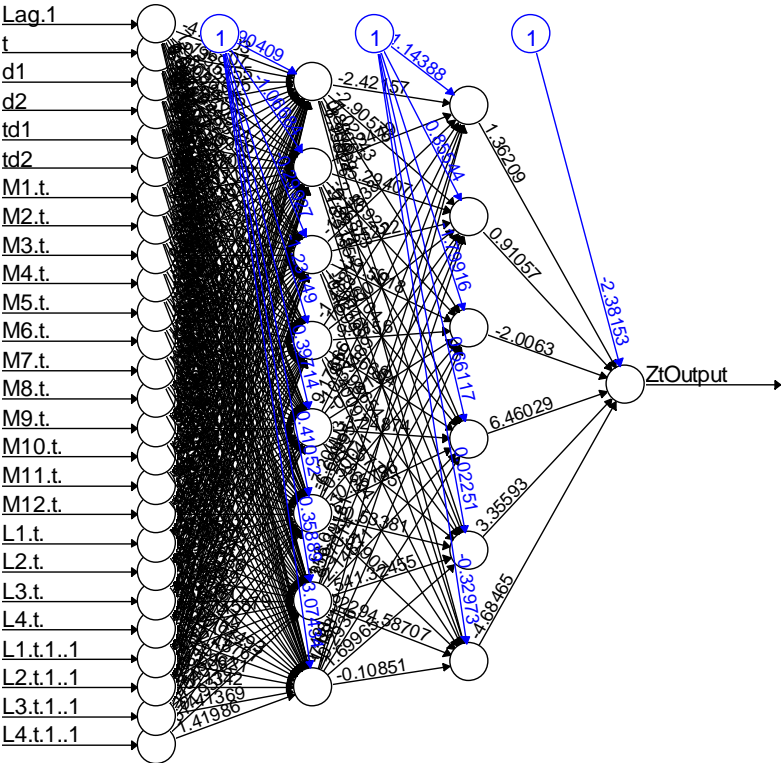
Lampiran 44. DNN-2 Inflow Rp 50.000,-



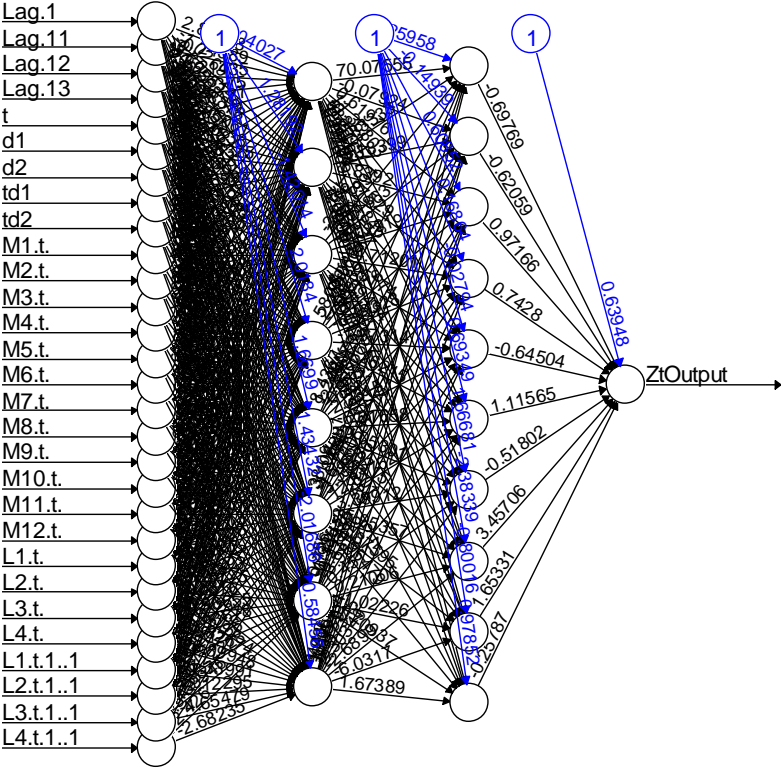
Lampiran 45. DNN-2 Inflow Rp 20.000,-



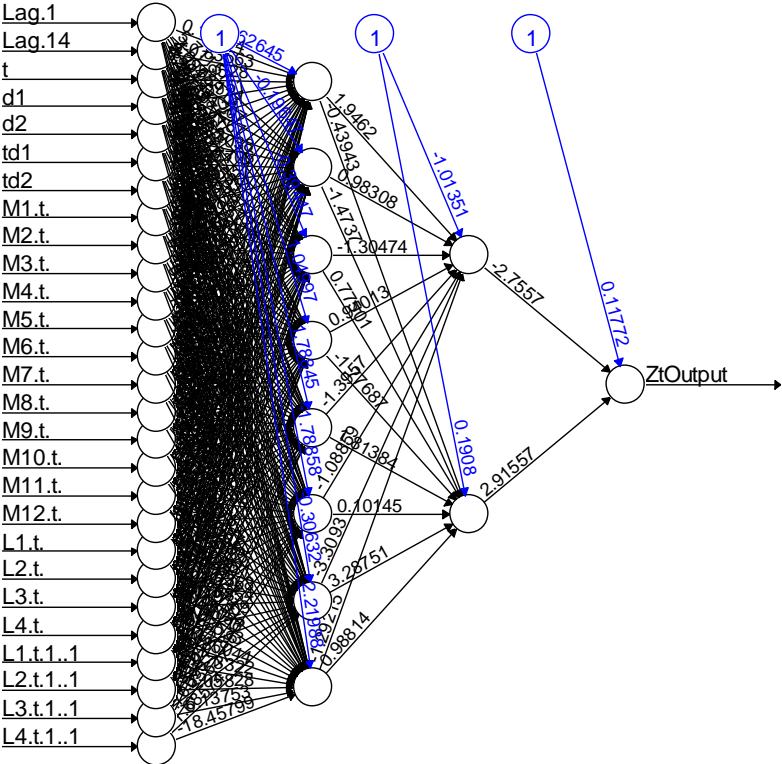
Lampiran 46. DNN-2 Inflow Rp 10.000,-



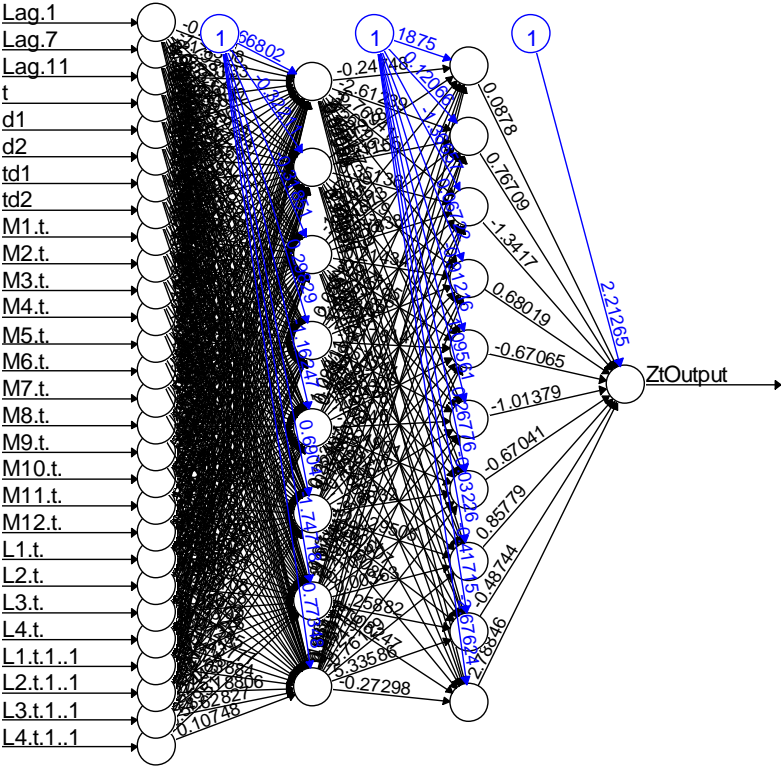
Lampiran 47. DNN-2 Inflow Rp 5.000,-



Lampiran 48. DNN-2 Inflow Rp 2.000,-



Lampiran 49. DNN-2 Inflow Rp 1.000,-



Lampiran 50. ADF Test & White Test Input DNN dari SSA-DNN

Uji Stasioneritas dalam *mean* dengan ADF Test

Data	Lag	Statistic	P	Data	Lag	Statistic	P
Noise Y1,t	5	-4.80	0.01	Noise Y8,t	5	-4.10	0.01
Noise Y2,t	5	-4.00	0.01	Noise Y9,t	5	-3.70	0.02
Noise Y3,t	5	-5.00	0.01	Noise Y10,t	5	-4.40	0.01
Noise Y4,t	5	-6.00	0.01	Noise Y11,t	5	-5.60	0.01
Noise Y5,t	5	-5.50	0.01	Noise Y12,t	5	-6.30	0.01
Noise Y6,t	4	-4.30	0.01	Noise Y13,t	4	-4.30	0.01
Noise Y7,t	5	-4.20	0.01	Noise Y14,t	5	-5.50	0.01

Uji Linieritas dengan lag *noise* Y_t menggunakan white test.

P value kurang dari 5% menyimpulkan bahwa hyimpulkan bahwa terdapat hubungan non-linier antara $Y_{i,t}$ dengan lag-nya.

Lag Noise	Ulangan ke-										% (Non-linier)
Y1,t	1	2	3	4	5	6	7	8	9	10	
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	100%
2	0.000	0.000	0.000	0.000	0.000	0.000	0.014	0.000	0.000	0.000	100%
3	0.000	0.034	0.018	0.001	0.000	0.001	0.000	0.001	0.350	0.001	90%
13	0.006	0.620	0.350	0.005	0.160	0.340	0.780	0.290	0.410	0.140	20%
Lag Noise	Ulangan ke-										% (Non-linier)
Y2,t	1	2	3	4	5	6	7	8	9	10	
15	0.190	0.710	0.450	0.750	0.800	0.630	0.270	0.780	0.410	0.680	0%
Lag Noise	Ulangan ke-										% (Non-linier)
Y3,t	1	2	3	4	5	6	7	8	9	10	
20	0.170	0.560	0.150	0.240	0.530	0.200	0.130	0.220	0.002	0.610	10%
24	0.980	0.200	0.220	0.590	0.160	0.260	0.880	0.120	0.098	0.150	0%
Lag Noise	Ulangan ke-										% (Non-linier)
Y4,t	1	2	3	4	5	6	7	8	9	10	
22	0.043	0.110	0.750	0.710	0.210	0.480	0.190	0.006	0.760	0.270	20%
Lag Noise	Ulangan ke-										% (Non-linier)
Y5,t	1	2	3	4	5	6	7	8	9	10	
1	0.582	0.440	0.240	0.820	0.007	0.130	0.480	0.420	0.148	0.088	10%
23	0.002	0.120	1.000	0.890	0.013	0.870	0.340	0.420	0.021	0.439	30%
Lag Noise	Ulangan ke-										% (Non-linier)
Y6,t	1	2	3	4	5	6	7	8	9	10	
11	0.889	0.069	0.510	0.600	0.003	0.028	0.341	0.784	0.183	0.510	20%
12	0.708	0.000	0.290	0.640	0.144	0.064	0.273	0.001	0.013	0.064	30%
13	0.015	0.001	0.190	0.000	0.015	0.093	0.000	0.213	0.013	0.000	70%
22	0.276	0.003	0.002	0.021	0.000	0.020	0.009	0.002	0.044	0.003	90%
24	0.002	0.330	0.000	0.004	0.003	0.145	0.000	0.779	0.490	0.000	60%

Lag Noise Y7,t	Ulangan ke-										% (Non- linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.013	0.004	0.005	0.005	0.004	0.011	0.006	0.013	0.003	0.005	100%
24	0.185	0.002	0.702	0.268	0.984	0.944	0.557	0.035	0.748	0.377	20%

Lag Noise Y8,t	Ulangan ke-										% (Non- linier)
	1	2	3	4	5	6	7	8	9	10	
16	0.950	0.110	0.440	0.130	0.501	0.830	0.416	0.900	0.003	0.363	10%
23	0.580	0.900	0.370	0.610	0.048	0.081	0.075	0.082	0.127	0.000	20%

Lag Noise Y9,t	Ulangan ke-										% (Non- linier)
	1	2	3	4	5	6	7	8	9	10	
15	0.140	0.860	0.625	0.881	0.540	0.330	0.950	0.153	0.286	0.570	0%
22	0.060	0.570	0.637	0.100	0.140	0.780	0.620	0.105	0.200	0.700	0%
24	0.310	0.230	0.013	0.047	0.170	0.510	0.280	0.031	0.049	0.800	40%

Lag Noise Y10,t	Ulangan ke-										% (Non- linier)
	1	2	3	4	5	6	7	8	9	10	
3	0.097	0.630	0.134	0.880	0.820	0.054	0.004	0.270	0.680	0.050	10%
12	0.101	0.270	0.614	0.900	0.640	0.021	0.343	0.920	0.960	0.887	10%
24	0.561	0.910	0.007	0.410	0.300	0.901	0.128	0.120	0.750	0.017	20%

Lag Noise Y11,t	Ulangan ke-										% (Non- linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.069	0.093	0.082	0.190	0.220	0.277	0.052	0.021	0.210	0.277	10%
11	0.015	0.946	0.251	0.810	0.110	0.055	0.345	0.121	0.660	0.068	10%
13	0.247	0.046	0.320	0.420	0.550	0.001	0.044	0.347	0.510	0.069	30%
24	0.812	0.250	0.716	0.230	0.120	0.855	0.239	0.443	0.150	0.626	0%

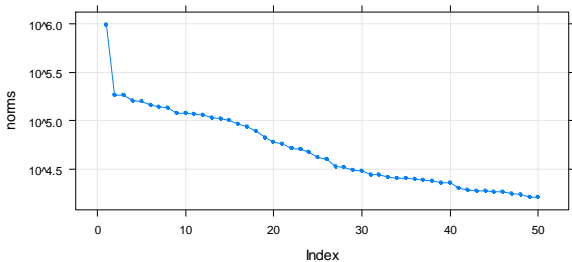
Lag Noise Y12,t	Ulangan ke-										% (Non- linier)
	1	2	3	4	5	6	7	8	9	10	
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	100%
11	0.065	0.060	0.370	0.210	0.083	0.200	0.850	0.824	0.054	0.094	0%
12	0.019	0.225	0.780	0.640	0.510	0.013	0.160	0.516	0.190	0.009	30%
13	0.039	0.676	0.870	0.930	0.680	0.550	0.950	0.024	0.140	0.210	20%
24	0.540	0.254	0.380	0.320	0.032	0.910	0.250	0.940	0.640	0.960	10%

Lag Noise Y13,t	Ulangan ke-										% (Non- linier)
	1	2	3	4	5	6	7	8	9	10	
5	0.728	0.770	0.819	0.140	0.520	0.680	0.034	0.150	0.006	0.198	20%
12	0.023	0.390	0.037	0.760	0.430	0.710	0.071	0.130	0.936	0.012	30%
14	0.059	0.120	0.474	0.930	0.150	0.400	0.060	0.910	0.033	0.117	10%

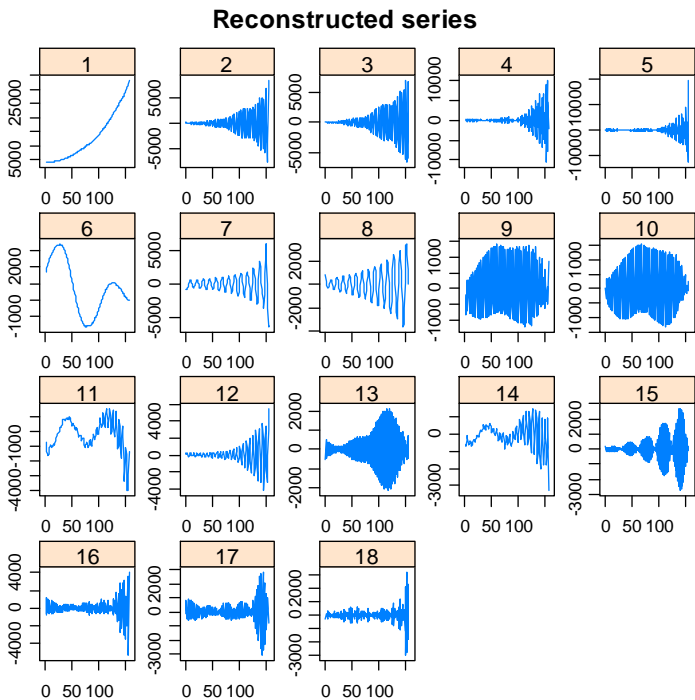
Lag Noise Y14,t	Ulangan ke-										% (Non- linier)
	1	2	3	4	5	6	7	8	9	10	
3	0.200	0.304	0.990	0.647	0.087	0.375	0.482	0.289	0.465	0.403	0%
4	0.960	0.057	0.420	0.945	0.072	0.598	0.002	0.333	0.302	0.564	10%
12	0.680	0.153	0.330	0.013	0.508	0.835	0.525	0.048	0.749	0.038	30%
13	0.970	0.117	0.230	0.262	0.598	0.002	0.064	0.175	0.082	0.205	10%

Lampiran 51. SSA-DNN *Outflow* Rp 100.000,-

A. Plot Eigen Value



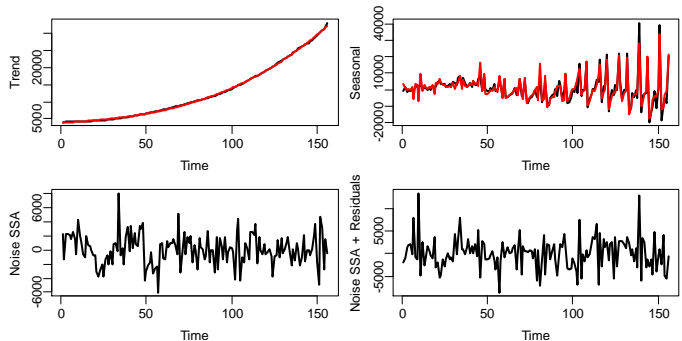
B. Rekonstruksi Individu



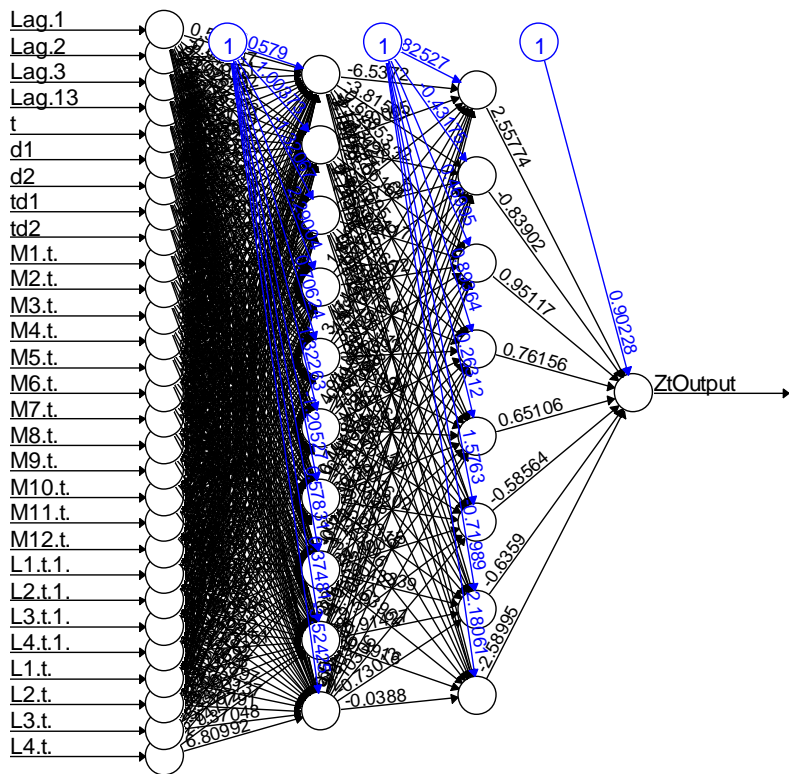
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	3.874e+03	5.595e+01	69.247	< 2e-16	***
t	2.599e+01	3.076e+00	8.449	2.21e-14	***
v2	3.202e-01	4.545e-02	7.044	6.06e-11	***
v3	4.365e-03	1.903e-04	22.932	< 2e-16	***
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	-214.01	34.78	-6.154	1.03e-08	***
d1	-10394.76	3553.82	-2.925	0.004120	**
d2	-5220.43	4318.40	-1.209	0.229085	
td1	186.00	46.10	4.035	9.66e-05	***
td2	-11.07	45.09	-0.245	0.806545	
M1.t.	13726.51	3869.29	3.548	0.000556	***
M2.t.	11640.38	3870.07	3.008	0.003207	**
M3.t.	9674.10	3870.86	2.499	0.013799	*
M4.t.	10372.85	3871.66	2.679	0.008416	**
M5.t.	9596.41	3872.50	2.478	0.014599	*
M6.t.	11312.31	3893.12	2.906	0.004364	**
M7.t.	2920.81	3907.77	0.747	0.456262	
M8.t.	14024.66	3880.98	3.614	0.000442	***
M9.t.	13234.94	3961.96	3.341	0.001115	**
M10.t.	3815.20	4220.24	0.904	0.367794	
M11.t.	1759.38	4081.40	0.431	0.667188	
M12.t.	6304.68	3881.30	1.624	0.106920	
L1.t.1.	10315.38	2759.82	3.738	0.000286	***
L2.t.1.	10017.22	2040.85	4.908	2.92e-06	***
L3.t.1.	-473.51	2282.82	-0.207	0.836030	
L4.t.1.	-2004.69	2076.73	-0.965	0.336330	
L1.t.	3174.61	2733.59	1.161	0.247810	
L2.t.	9815.41	1995.24	4.919	2.79e-06	***
L3.t.	21045.43	2323.34	9.058	2.99e-15	***
L4.t.	16590.75	2103.95	7.886	1.62e-12	***
tm1.t.	NA	NA	NA	NA	
tm2.t.	45.77	30.54	1.499	0.136623	
tm3.t.	105.28	30.55	3.446	0.000785	***
tm4.t.	117.36	30.57	3.840	0.000198	***
tm5.t.	122.56	30.58	4.007	0.000107	***
tm6.t.	141.04	31.77	4.439	2.02e-05	***
tm7.t.	277.81	33.00	8.419	9.45e-14	***
tm8.t.	43.68	31.31	1.395	0.165635	
tm9.t.	58.79	30.97	1.898	0.060072	.
tm10.t.	167.19	33.23	5.032	1.73e-06	***
tm11.t.	196.08	32.49	6.035	1.82e-08	***
tm12.t.	312.35	30.85	10.124	< 2e-16	***

D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy

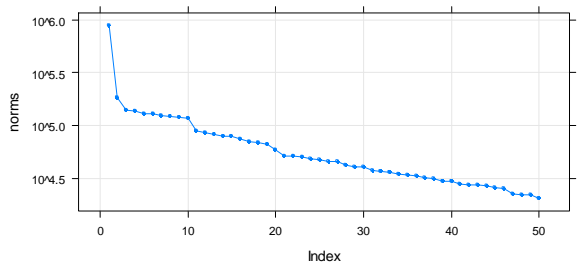


E. Arsitektur DNN Terbaik

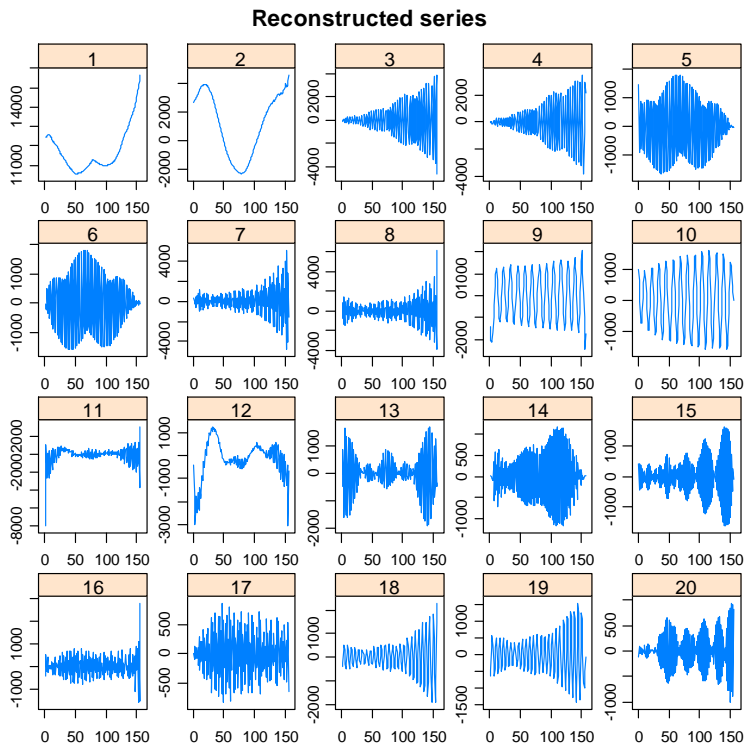


Lampiran 52. SSA-DNN *Outflow* Rp 50.000,-

A. Plot Eigen Value



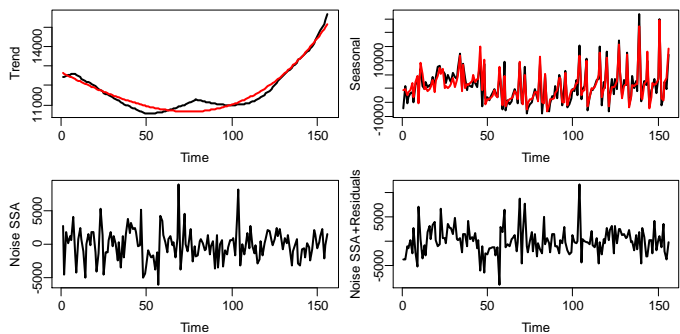
B. Rekonstruksi Individu



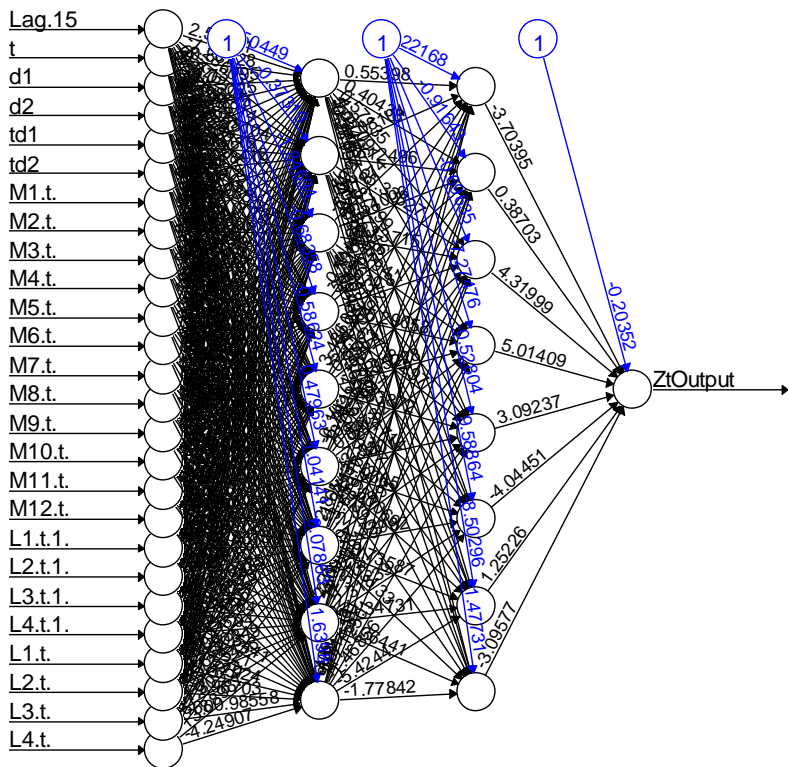
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.267e+04	8.331e+01	152.062	< 2e-16 ***
t	-4.239e+01	4.581e+00	-9.254	< 2e-16 ***
v2	5.454e-02	6.768e-02	0.806	0.422
v3	2.037e-03	2.834e-04	7.186	2.81e-11 ***
Regresi <i>dummy</i>				
	Estimate	Std. Error	t value	Pr(> t)
t	-21.887	25.277	-0.866	0.388289
d1	3316.134	2583.161	1.284	0.201703
d2	2285.213	3138.906	0.728	0.468014
td1	46.701	33.509	1.394	0.165991
td2	-57.856	32.771	-1.765	0.080031 .
M1.t.	-4003.670	2812.468	-1.424	0.157174
M2.t.	-3620.183	2813.035	-1.287	0.200595
M3.t.	-6020.728	2813.603	-2.140	0.034390 *
M4.t.	-5043.213	2814.185	-1.792	0.075642 .
M5.t.	-4397.686	2814.797	-1.562	0.120840
M6.t.	-1896.686	2829.783	-0.670	0.503981
M7.t.	-7717.079	2840.437	-2.717	0.007565 **
M8.t.	-1396.023	2820.959	-0.495	0.621593
M9.t.	970.418	2879.821	0.337	0.736727
M10.t.	-8170.756	3067.560	-2.664	0.008793 **
M11.t.	-10268.031	2966.639	-3.461	0.000746 ***
M12.t.	305.850	2821.192	0.108	0.913850
L1.t.1.	12784.077	2006.026	6.373	3.58e-09 ***
L2.t.1.	9603.781	1483.432	6.474	2.18e-09 ***
L3.t.1.	8.085	1659.309	0.005	0.996120
L4.t.1.	-1621.316	1509.507	-1.074	0.284946
L1.t.	1026.861	1986.957	0.517	0.606246
L2.t.	9059.805	1450.278	6.247	6.60e-09 ***
L3.t.	15784.646	1688.764	9.347	6.18e-16 ***
L4.t.	14788.917	1529.297	9.670	< 2e-16 ***
tM1.t.	NA	NA	NA	NA
tM2.t.	24.328	22.202	1.096	0.275371
tM3.t.	75.736	22.208	3.410	0.000884 ***
tM4.t.	64.667	22.217	2.911	0.004300 **
tM5.t.	64.339	22.230	2.894	0.004516 **
tM6.t.	57.734	23.092	2.500	0.013763 *
tM7.t.	131.389	23.985	5.478	2.41e-07 ***
tM8.t.	7.836	22.761	0.344	0.731259
tM9.t.	-12.716	22.512	-0.565	0.573232
tM10.t.	90.124	24.153	3.731	0.000292 ***
tM11.t.	111.028	23.618	4.701	6.97e-06 ***
tM12.t.	112.376	22.425	5.011	1.89e-06 ***

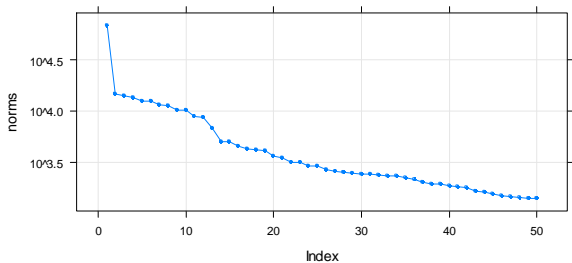
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



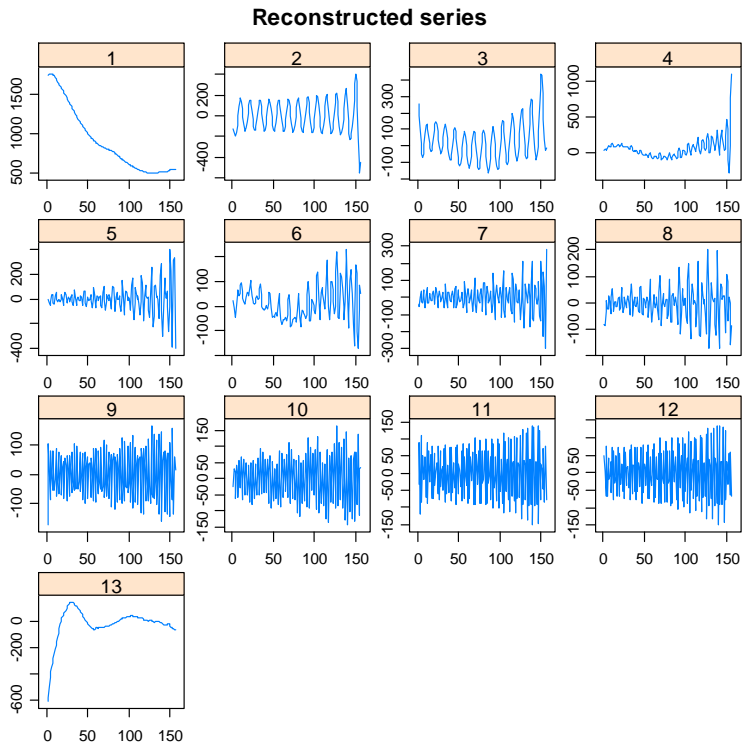
E. Arsitektur DNN Terbaik



Lampiran 53. SSA-DNN Outflow Rp 20.000,-
A. Plot Eigen Value



B. Rekonstruksi Individu

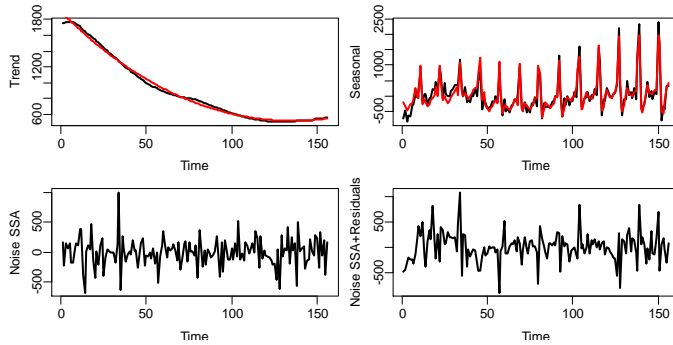


C. Estimasi parameter regresi polinomial & regresi *dummy*

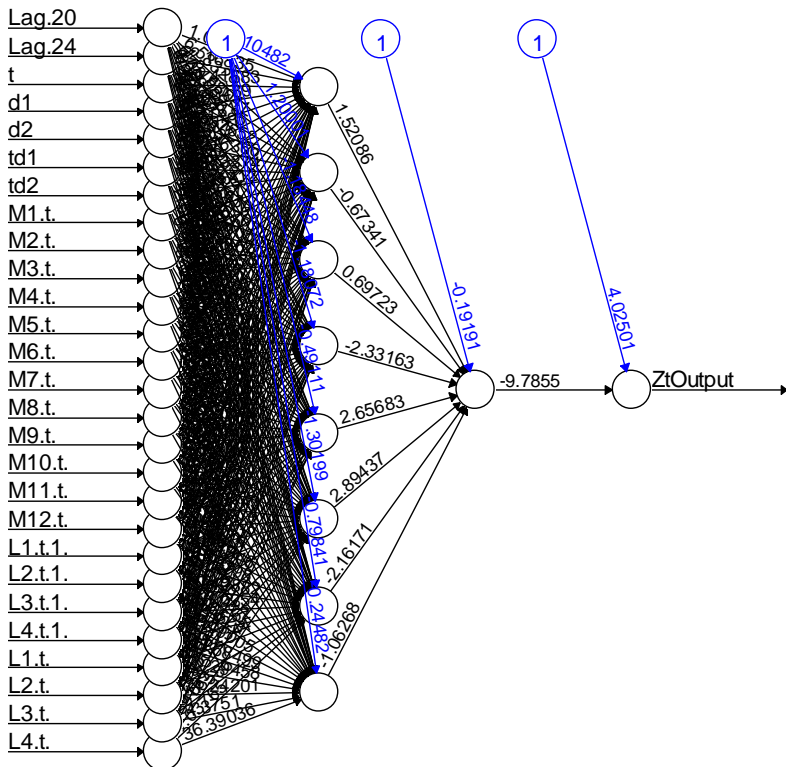
Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	1.895e+03	9.550e+00	198.466	<2e-16	***
t	-2.141e+01	5.251e-01	-40.767	<2e-16	***
v2	9.129e-02	7.759e-03	11.767	<2e-16	***
v3	-6.047e-05	3.249e-05	-1.861	0.0646	.

Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	2.7819	2.0491	1.358	0.177123	
d1	114.0350	209.4019	0.545	0.587057	
d2	-15.5970	254.4529	-0.061	0.951225	
td1	5.2423	2.7164	1.930	0.055981	.
td2	-2.0596	2.6566	-0.775	0.439687	
M1.t.	-315.8055	227.9904	-1.385	0.168570	
M2.t.	-443.4752	228.0364	-1.945	0.054144	.
M3.t.	-595.8052	228.0824	-2.612	0.010144	*
M4.t.	-536.1939	228.1296	-2.350	0.020385	*
M5.t.	-414.5486	228.1793	-1.817	0.071749	.
M6.t.	-405.6579	229.3941	-1.768	0.079537	.
M7.t.	-273.8187	230.2577	-1.189	0.236715	
M8.t.	146.0549	228.6787	0.639	0.524242	
M9.t.	-175.0447	233.4504	-0.750	0.454833	
M10.t.	-911.6836	248.6692	-3.666	0.000368	***
M11.t.	-832.3899	240.4882	-3.461	0.000746	***
M12.t.	-510.0323	228.6976	-2.230	0.027596	*
L1.t.1.	1543.5805	162.6169	9.492	2.79e-16	***
L2.t.1.	1571.5931	120.2532	13.069	< 2e-16	***
L3.t.1.	1353.0080	134.5106	10.059	< 2e-16	***
L4.t.1.	617.1635	122.3670	5.044	1.64e-06	***
L1.t.	403.3656	161.0711	2.504	0.013613	*
L2.t.	913.6429	117.5656	7.771	2.95e-12	***
L3.t.	1298.2255	136.8983	9.483	2.93e-16	***
L4.t.	1570.0233	123.9712	12.664	< 2e-16	***
tm1.t.	NA	NA	NA	NA	
tm2.t.	0.6655	1.7998	0.370	0.712198	
tm3.t.	1.6387	1.8002	0.910	0.364497	
tm4.t.	2.2425	1.8010	1.245	0.215506	
tm5.t.	1.3622	1.8021	0.756	0.451185	
tm6.t.	3.9480	1.8719	2.109	0.037017	*
tm7.t.	2.1015	1.9443	1.081	0.281941	
tm8.t.	-6.9494	1.8451	-3.766	0.000258	***
tm9.t.	-5.2373	1.8249	-2.870	0.004854	**
tm10.t.	2.0136	1.9579	1.028	0.305820	
tm11.t.	4.2357	1.9146	2.212	0.028834	*
tm12.t.	3.1867	1.8179	1.753	0.082156	.

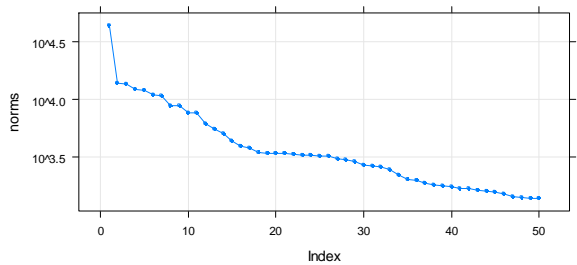
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



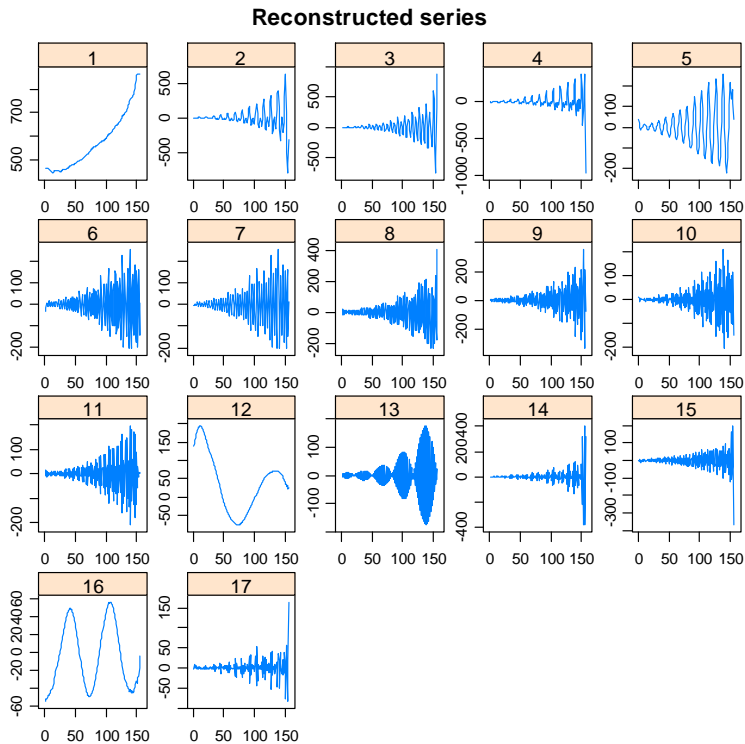
E. Arsitektur DNN Terbaik



Lampiran 54. SSA-DNN *Outflow* Rp 10.000,-
A. Plot Eigen Value



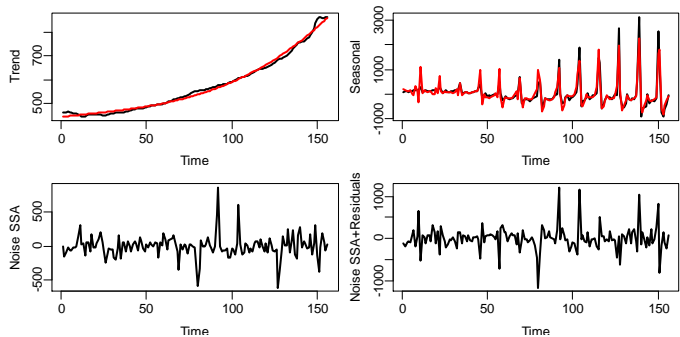
B. Rekonstruksi Individu



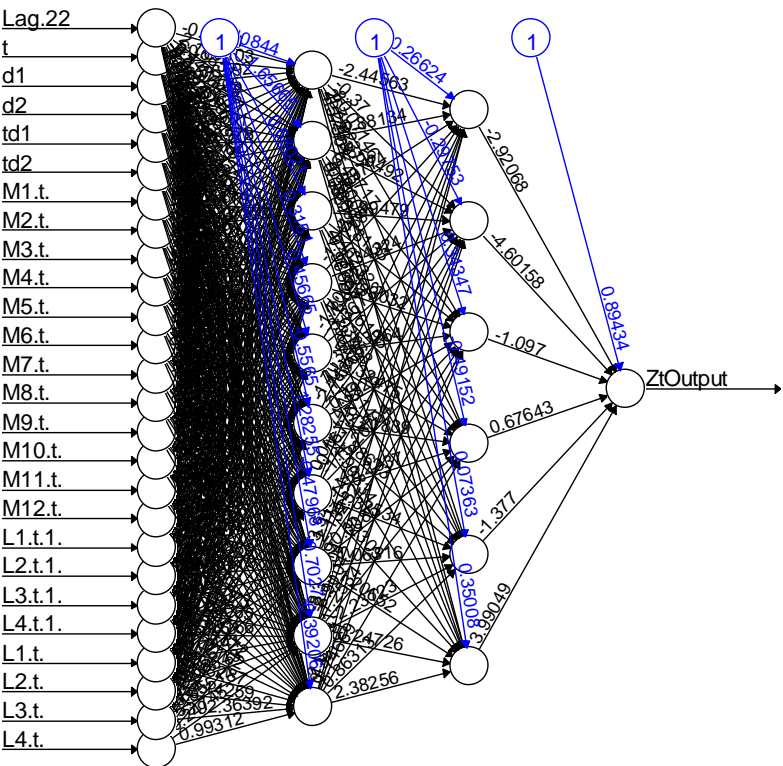
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	4.442e+02	3.406e+00	130.416	< 2e-16	***
t	7.380e-01	1.873e-01	3.940	0.000124	***
v2	-2.020e-03	2.767e-03	-0.730	0.466617	
v3	9.273e-05	1.159e-05	8.003	2.91e-13	***
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	-3.7181	2.4705	-1.505	0.134945	
d1	-110.5694	252.4658	-0.438	0.662204	
d2	-288.0013	306.7816	-0.939	0.349728	
td1	0.7994	3.2750	0.244	0.807576	
td2	1.2210	3.2029	0.381	0.703712	
M1.t.	292.6493	274.8771	1.065	0.289170	
M2.t.	286.0892	274.9325	1.041	0.300163	
M3.t.	243.0915	274.9880	0.884	0.378460	
M4.t.	244.4345	275.0449	0.889	0.375938	
M5.t.	255.7679	275.1048	0.930	0.354387	
M6.t.	71.4908	276.5694	0.258	0.796471	
M7.t.	2.4406	277.6106	0.009	0.993000	
M8.t.	490.7137	275.7070	1.780	0.077634	.
M9.t.	269.5205	281.4599	0.958	0.340200	
M10.t.	-573.4842	299.8085	-1.913	0.058152	.
M11.t.	-417.2134	289.9450	-1.439	0.152771	
M12.t.	140.7431	275.7297	0.510	0.610682	
L1.t.1.	1097.2379	196.0593	5.596	1.41e-07	***
L2.t.1.	1384.4089	144.9835	9.549	< 2e-16	***
L3.t.1.	1264.9198	162.1729	7.800	2.54e-12	***
L4.t.1.	318.8173	147.5319	2.161	0.032682	*
L1.t.	414.7844	194.1957	2.136	0.034718	*
L2.t.	597.9562	141.7432	4.219	4.80e-05	***
L3.t.	1078.3509	165.0517	6.533	1.63e-09	***
L4.t.	1610.0132	149.4662	10.772	< 2e-16	***
tM1.t.	NA	NA	NA	NA	
tM2.t.	-0.2315	2.1699	-0.107	0.915202	
tM3.t.	0.2201	2.1705	0.101	0.919398	
tM4.t.	0.4963	2.1714	0.229	0.819581	
tM5.t.	1.1665	2.1726	0.537	0.592322	
tM6.t.	5.6866	2.2569	2.520	0.013060	*
tM7.t.	8.4957	2.3442	3.624	0.000426	***
tM8.t.	-3.3639	2.2246	-1.512	0.133132	
tM9.t.	-3.4355	2.2002	-1.561	0.121045	
tM10.t.	4.4263	2.3606	1.875	0.063212	.
tM11.t.	4.8813	2.3083	2.115	0.036526	*
tM12.t.	2.2882	2.1917	1.044	0.298563	

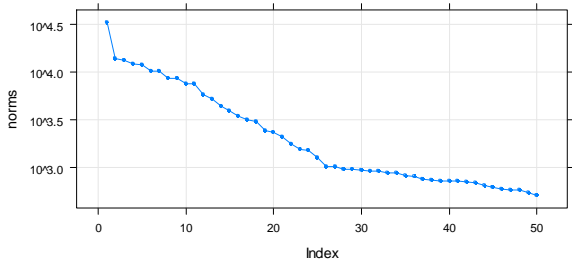
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



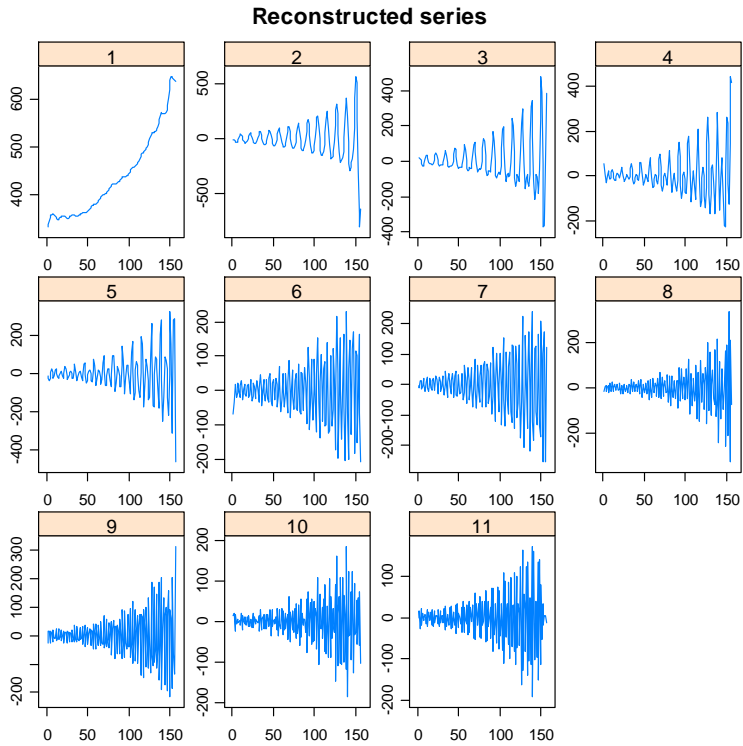
E. Arsitektur DNN Terbaik



Lampiran 55. SSA-DNN Outflow Rp 5.000,-
A. Plot Eigen Value



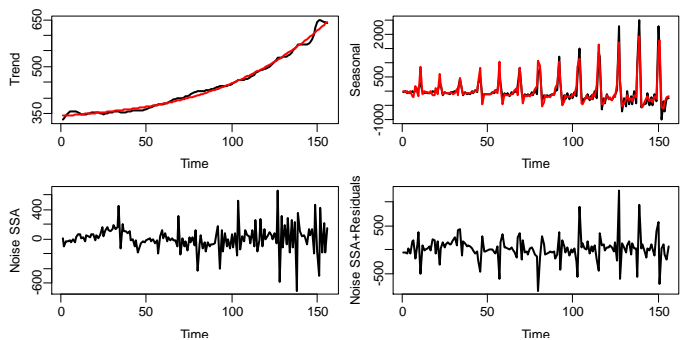
B. Rekonstruksi Individu



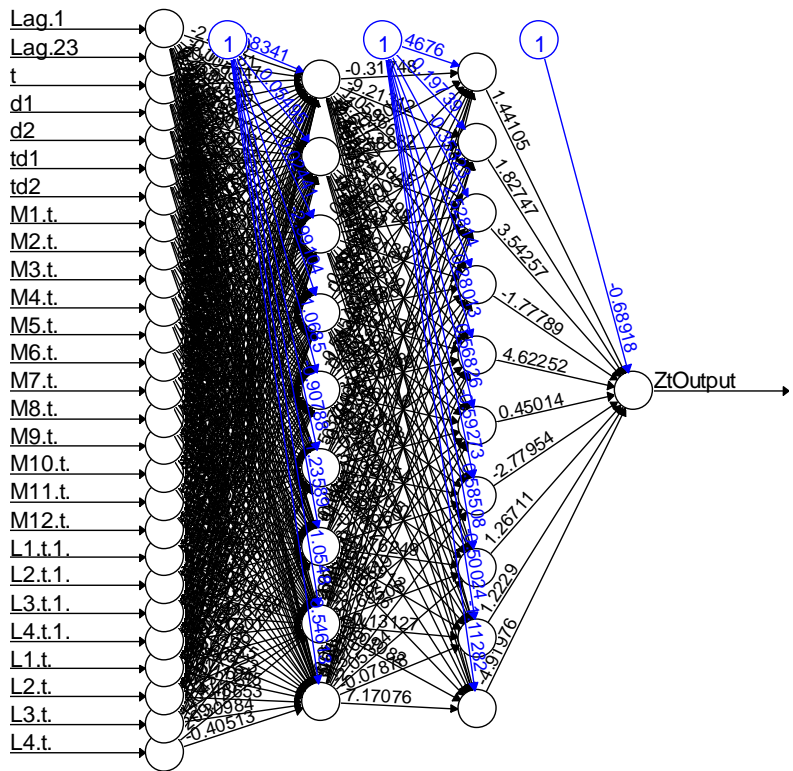
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	3.430e+02	2.545e+00	134.782	< 2e-16	***
t	4.571e-01	1.399e-01	3.267	0.00134	**
v2	-5.028e-04	2.067e-03	-0.243	0.80816	
v3	6.354e-05	8.657e-06	7.340	1.2e-11	***
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	-3.41219	2.08044	-1.640	0.10360	
d1	-104.61541	212.60785	-0.492	0.62358	
d2	-69.36275	258.34856	-0.268	0.78879	
td1	1.25776	2.75796	0.456	0.64918	
td2	0.41768	2.69724	0.155	0.87720	
M1.t.	113.80158	231.48099	0.492	0.62388	
M2.t.	67.36064	231.52765	0.291	0.77160	
M3.t.	94.07862	231.57438	0.406	0.68528	
M4.t.	20.03823	231.62233	0.087	0.93120	
M5.t.	116.29353	231.67271	0.502	0.61661	
M6.t.	-113.69658	232.90614	-0.488	0.62632	
M7.t.	-170.31559	233.78296	-0.729	0.46771	
M8.t.	301.96589	232.17984	1.301	0.19590	
M9.t.	44.82327	237.02453	0.189	0.85033	
M10.t.	-655.61796	252.47639	-2.597	0.01059	*
M11.t.	-522.24610	244.17006	-2.139	0.03447	*
M12.t.	-21.78192	232.19900	-0.094	0.92542	
L1.t.1.	1121.60608	165.10655	6.793	4.49e-10	***
L2.t.1.	1289.82488	122.09429	10.564	< 2e-16	***
L3.t.1.	1275.57251	136.56993	9.340	6.42e-16	***
L4.t.1.	539.87028	124.24040	4.345	2.93e-05	***
L1.t.	547.81070	163.53710	3.350	0.00108	**
L2.t.	790.98782	119.36559	6.627	1.03e-09	***
L3.t.	1235.66568	138.99420	8.890	7.45e-15	***
L4.t.	1455.51218	125.86926	11.564	< 2e-16	***
tm1.t.	NA	NA	NA	NA	
tm2.t.	0.91912	1.82734	0.503	0.61590	
tm3.t.	0.08626	1.82780	0.047	0.96244	
tm4.t.	2.11984	1.82857	1.159	0.24864	
tm5.t.	0.83951	1.82964	0.459	0.64718	
tm6.t.	6.17725	1.90058	3.250	0.00150	**
tm7.t.	8.20950	1.97408	4.159	6.04e-05	***
tm8.t.	-2.39134	1.87339	-1.276	0.20425	
tm9.t.	-1.01886	1.85283	-0.550	0.58342	
tm10.t.	5.16207	1.98791	2.597	0.01059	*
tm11.t.	5.31000	1.94388	2.732	0.00725	**
tm12.t.	2.37759	1.84568	1.288	0.20016	

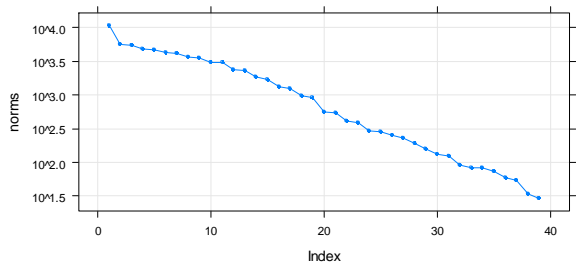
D. Rekonstruksi agregat, hasil *Trend Analysis & Regresi Dummy*



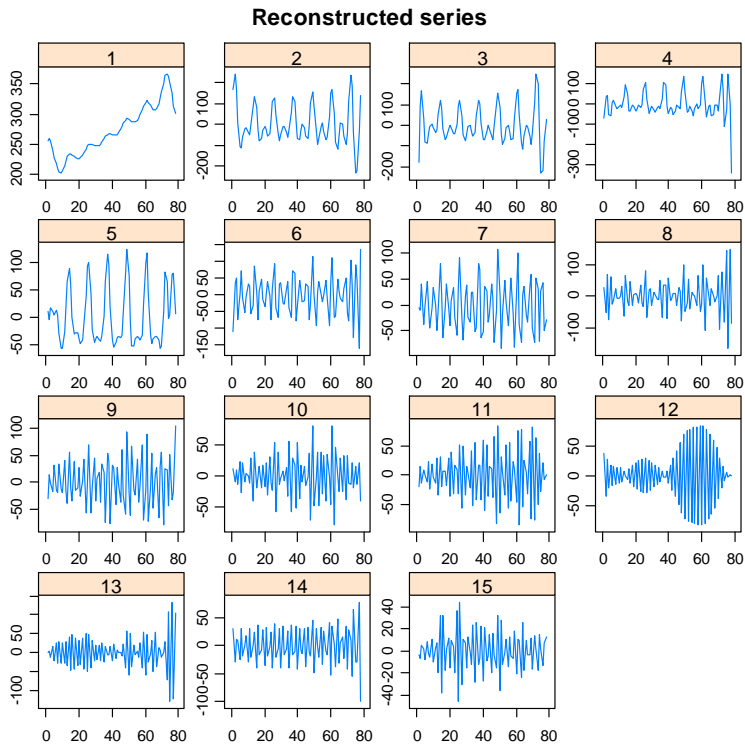
E. Arsitektur DNN Terbaik



Lampiran 56. SSA-DNN *Outflow* Rp 2.000,-
A. Plot Eigen Value



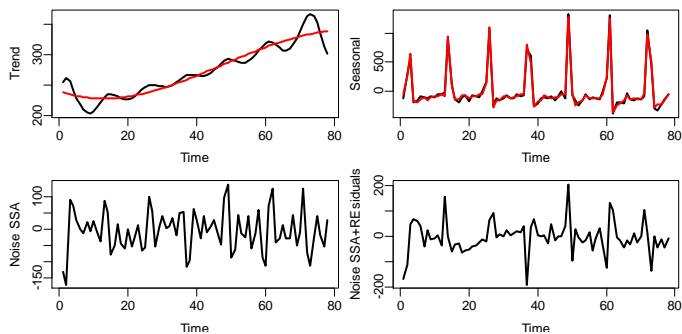
B. Rekonstruksi Individu



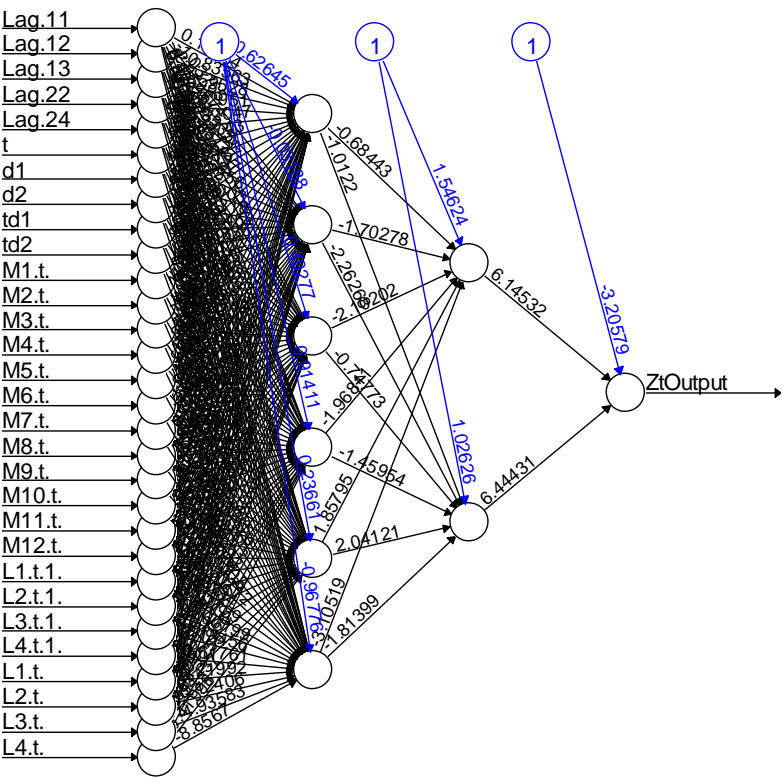
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	2.407e+02	6.049e+00	39.797	< 2e-16	***
t	-2.107e+00	6.589e-01	-3.198	0.00204	**
v2	9.490e-02	1.931e-02	4.914	5.21e-06	***
v3	-6.648e-04	1.608e-04	-4.135	9.26e-05	***
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	-1.372e+00	1.081e+00	-1.269	0.21078	
d1	NA	NA	NA	NA	
d2	-4.078e+02	2.398e+02	-1.700	0.09582	.
td1	NA	NA	NA	NA	
td2	4.381e+00	2.700e+00	1.623	0.11152	
M1.t.	4.791e+01	1.272e+02	0.377	0.70814	
M2.t.	-1.137e+02	1.278e+02	-0.889	0.37844	
M3.t.	1.679e+01	1.286e+02	0.131	0.89670	
M4.t.	3.134e+01	1.296e+02	0.242	0.80995	
M5.t.	-9.792e+01	1.306e+02	-0.750	0.45732	
M6.t.	-1.190e+01	1.600e+02	-0.074	0.94101	
M7.t.	5.395e+02	1.318e+02	4.093	0.00017	***
M8.t.	-1.238e+03	1.709e+02	-7.245	3.91e-09	***
M9.t.	-3.995e+02	1.408e+02	-2.837	0.00675	**
M10.t.	-3.385e+01	1.091e+02	-0.310	0.75777	
M11.t.	-1.304e+02	1.097e+02	-1.189	0.24073	
M12.t.	-1.496e+02	1.106e+02	-1.352	0.18284	
L1.t.1.	NA	NA	NA	NA	
L2.t.1.	1.623e+03	6.251e+01	25.962	< 2e-16	***
L3.t.1.	1.063e+03	5.634e+01	18.876	< 2e-16	***
L4.t.1.	3.112e+02	4.974e+01	6.256	1.19e-07	***
L1.t.	NA	NA	NA	NA	
L2.t.	3.828e+02	4.943e+01	7.743	7.09e-10	***
L3.t.	1.015e+03	5.475e+01	18.537	< 2e-16	***
L4.t.	1.696e+03	6.197e+01	27.362	< 2e-16	***
tM1.t.	NA	NA	NA	NA	
tM2.t.	1.226e+00	1.476e+00	0.831	0.41050	
tM3.t.	2.479e-01	1.477e+00	0.168	0.86745	
tM4.t.	1.929e-01	1.479e+00	0.130	0.89678	
tM5.t.	1.699e+00	1.481e+00	1.147	0.25730	
tM6.t.	9.761e-01	1.734e+00	0.563	0.57619	
tM7.t.	-5.544e+00	1.587e+00	-3.494	0.00106	**
tM8.t.	7.554e+00	1.596e+00	4.734	2.13e-05	***
tM9.t.	2.455e+00	1.491e+00	1.647	0.10645	
tM10.t.	5.089e-02	1.332e+00	0.038	0.96970	
tM11.t.	1.098e+00	1.334e+00	0.823	0.41482	
tM12.t.	1.925e+00	1.337e+00	1.439	0.15686	

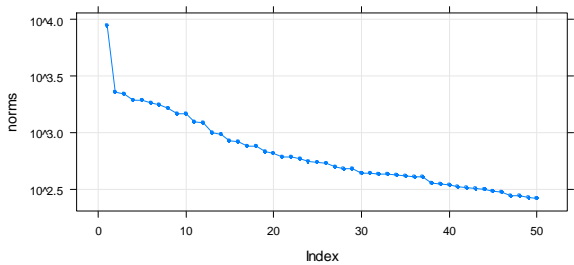
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



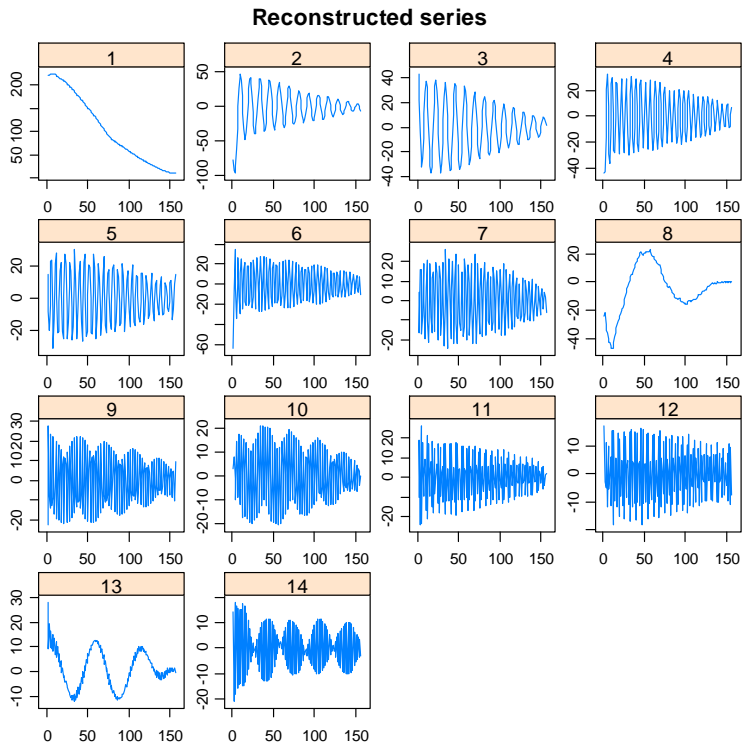
E. Arsitektur DNN Terbaik



Lampiran 57. SSA-DNN Outflow Rp 1.000,-
A. Plot Eigen Value



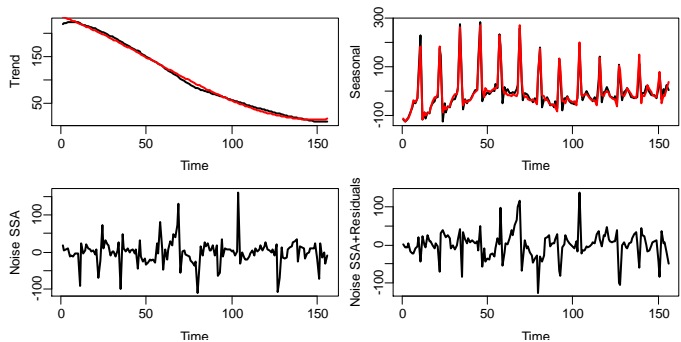
B. Rekonstruksi Individu



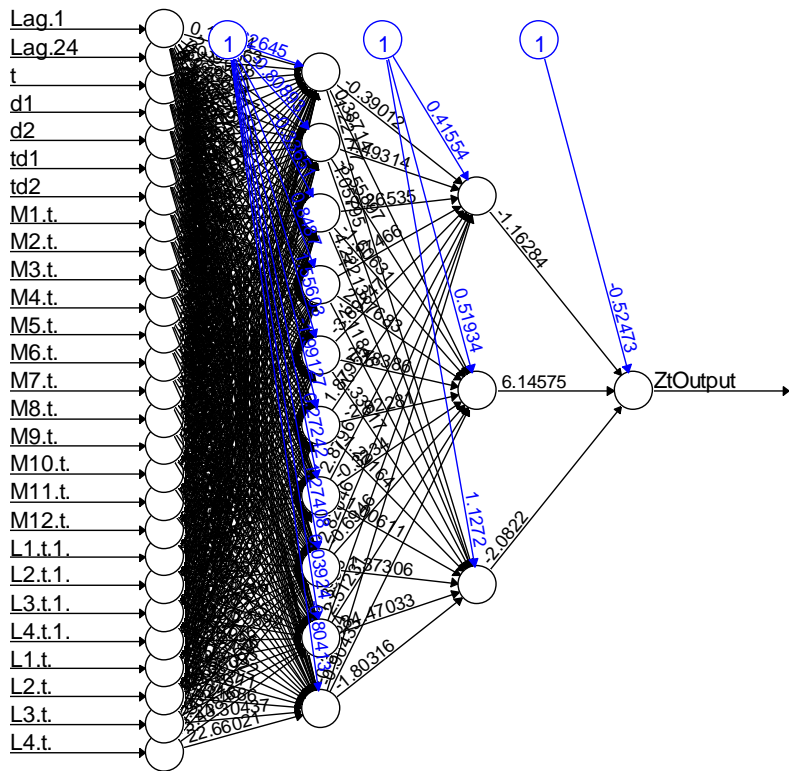
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	2.368e+02	1.469e+00	161.20	<2e-16	***
t	-1.323e+00	8.077e-02	-16.38	<2e-16	***
v2	-1.236e-02	1.193e-03	-10.35	<2e-16	***
v3	7.572e-05	4.997e-06	15.15	<2e-16	***
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	0.68707	0.19609	3.504	0.000645	***
d1	-24.08696	20.03910	-1.202	0.231731	
d2	171.55425	24.35034	7.045	1.26e-10	***
td1	1.48683	0.25995	5.720	7.98e-08	***
td2	-2.03670	0.25423	-8.011	8.32e-13	***
M1.t.	-92.31332	21.81797	-4.231	4.57e-05	***
M2.t.	-105.97234	21.82237	-4.856	3.65e-06	***
M3.t.	-96.14250	21.82677	-4.405	2.32e-05	***
M4.t.	-94.91963	21.83129	-4.348	2.90e-05	***
M5.t.	-76.03416	21.83604	-3.482	0.000695	***
M6.t.	-42.81922	21.95230	-1.951	0.053441	.
M7.t.	-18.47420	22.03494	-0.838	0.403470	
M8.t.	-32.56607	22.88384	-1.488	0.139339	
M9.t.	-17.95440	22.34047	-0.804	0.423176	
M10.t.	-18.89299	23.79687	-0.794	0.428805	
M11.t.	-59.67541	23.01396	-2.593	0.010697	*
M12.t.	-122.74091	21.88564	-5.608	1.33e-07	***
L1.t.1.	255.45489	15.56192	16.415	< 2e-16	***
L2.t.1.	191.75378	11.50785	16.663	< 2e-16	***
L3.t.1.	101.36060	12.87224	7.874	1.72e-12	***
L4.t.1.	67.49362	11.71013	5.764	6.51e-08	***
L1.t.	33.00194	15.41400	2.141	0.034294	*
L2.t.	135.96525	11.25066	12.085	< 2e-16	***
L3.t.	184.01840	13.10074	14.046	< 2e-16	***
L4.t.	244.64654	11.86366	20.622	< 2e-16	***
tm1.t.	NA	NA	NA	NA	
tm2.t.	0.00041	0.17223	0.002	0.998104	
tm3.t.	-0.14528	0.17228	-0.843	0.400753	
tm4.t.	-0.18211	0.17235	-1.057	0.292811	
tm5.t.	-0.34166	0.17245	-1.981	0.049854	*
tm6.t.	-0.84015	0.17914	-4.690	7.30e-06	***
tm7.t.	-1.25107	0.18607	-6.724	6.35e-10	***
tm8.t.	-0.80224	0.17657	-4.543	1.33e-05	***
tm9.t.	-0.69113	0.17464	-3.958	0.000129	***
tm10.t.	-0.70550	0.18737	-3.765	0.000259	***
tm11.t.	-0.26385	0.18322	-1.440	0.152446	
tm12.t.	0.32879	0.17396	1.890	0.061167	

D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy

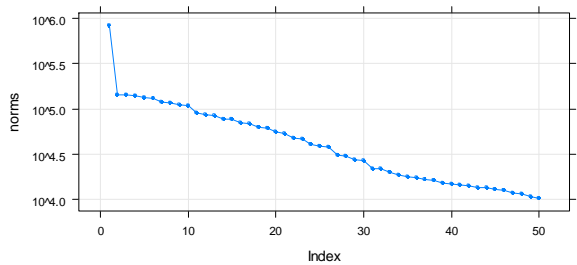


E. Arsitektur DNN Terbaik

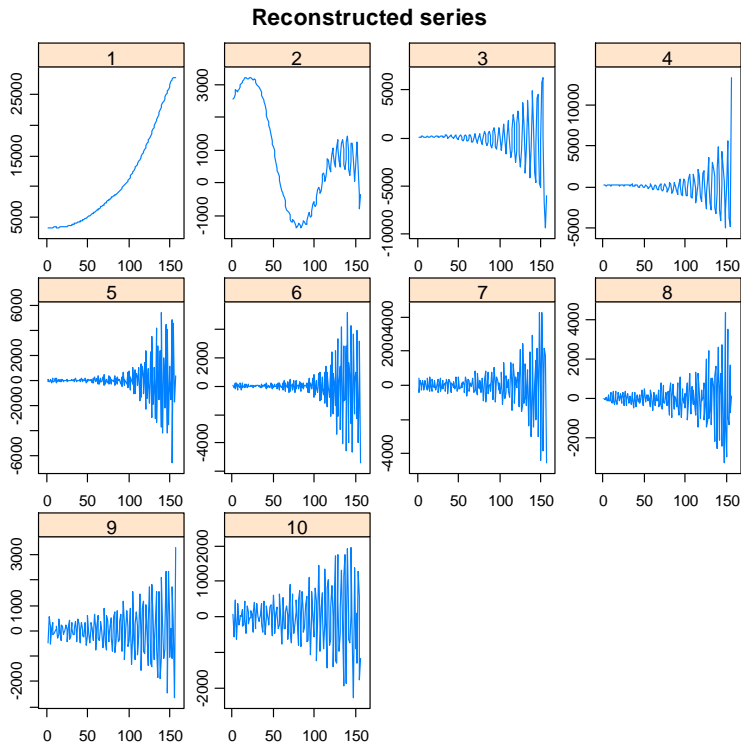


Lampiran 58. SSA-DNN Inflow Rp 100.000,-

A. Plot Eigen Value



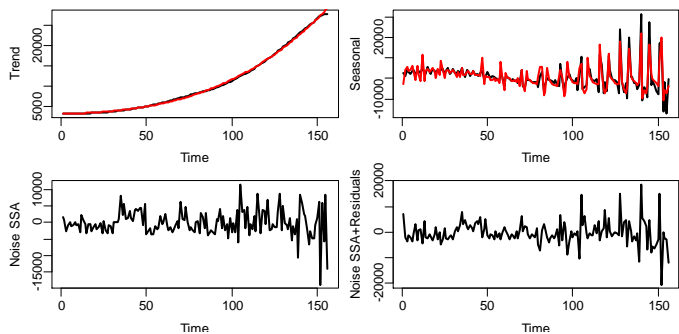
B. Rekonstruksi Individu



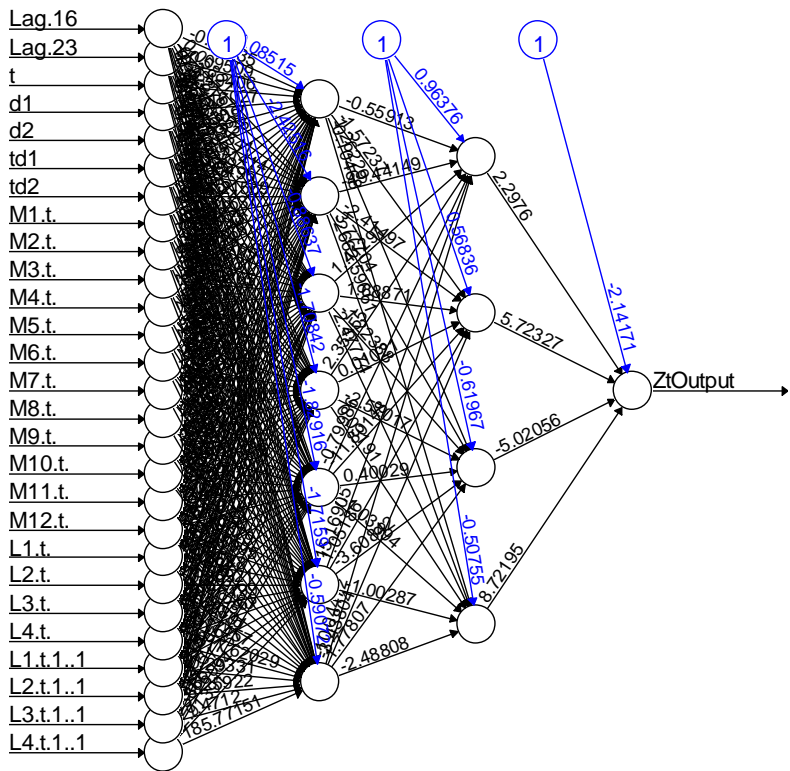
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	3.214e+03	8.064e+01	39.860	< 2e-16	***
t	1.239e+01	4.434e+00	2.794	0.005880	**
v2	2.528e-01	6.551e-02	3.858	0.000168	***
v3	4.679e-03	2.743e-04	17.056	< 2e-16	***
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	175.11	34.51	5.074	1.44e-06	***
d1	6165.75	3526.57	1.748	0.08296	.
d2	4813.00	4283.32	1.124	0.26340	
td1	-47.61	45.76	-1.040	0.30024	
td2	-66.27	44.74	-1.481	0.14114	
M1.t.	-8978.15	3839.36	-2.338	0.02102	*
M2.t.	-2189.33	3840.14	-0.570	0.56966	
M3.t.	-608.37	3840.91	-0.158	0.87441	
M4.t.	-2235.30	3841.71	-0.582	0.56176	
M5.t.	-1690.04	3842.55	-0.440	0.66086	
M6.t.	329.19	3843.45	0.086	0.93189	
M7.t.	-1991.10	3865.19	-0.515	0.60741	
M8.t.	-7231.61	3880.26	-1.864	0.06481	.
M9.t.	962.43	3852.26	0.250	0.80314	
M10.t.	-3148.75	3933.26	-0.801	0.42498	
M11.t.	-8169.39	4193.23	-1.948	0.05372	.
M12.t.	-8294.75	4054.32	-2.046	0.04295	*
L1.t.	8419.49	2738.79	3.074	0.00261	**
L2.t.	11279.61	2025.71	5.568	1.60e-07	***
L3.t.	11029.59	2265.66	4.868	3.47e-06	***
L4.t.	1033.55	2061.16	0.501	0.61698	
L1.t.1..1	4507.50	2712.76	1.662	0.09920	.
L2.t.1..1	6678.34	1980.47	3.372	0.00100	**
L3.t.1..1	10642.56	2305.92	4.615	9.91e-06	***
L4.t.1..1	14017.27	2088.27	6.712	6.72e-10	***
tM1.t.	NA	NA	NA	NA	
tM2.t.	-145.71	30.32	-4.806	4.50e-06	***
tM3.t.	-173.32	30.32	-5.716	8.12e-08	***
tM4.t.	-169.22	30.34	-5.578	1.53e-07	***
tM5.t.	-179.94	30.35	-5.928	3.02e-08	***
tM6.t.	-201.08	30.38	-6.620	1.07e-09	***
tM7.t.	-180.89	31.56	-5.732	7.53e-08	***
tM8.t.	-67.58	32.78	-2.062	0.04140	*
tM9.t.	-249.65	31.11	-8.023	7.80e-13	***
tM10.t.	-191.97	30.78	-6.237	6.93e-09	***
tM11.t.	-167.38	33.02	-5.069	1.47e-06	***
tM12.t.	-144.72	32.30	-4.481	1.71e-05	***

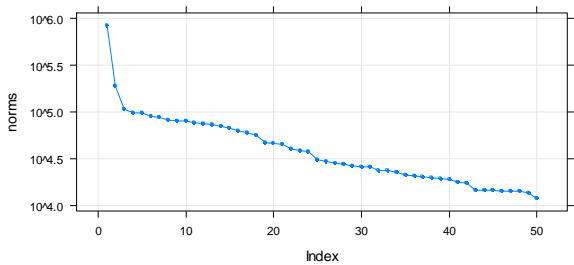
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



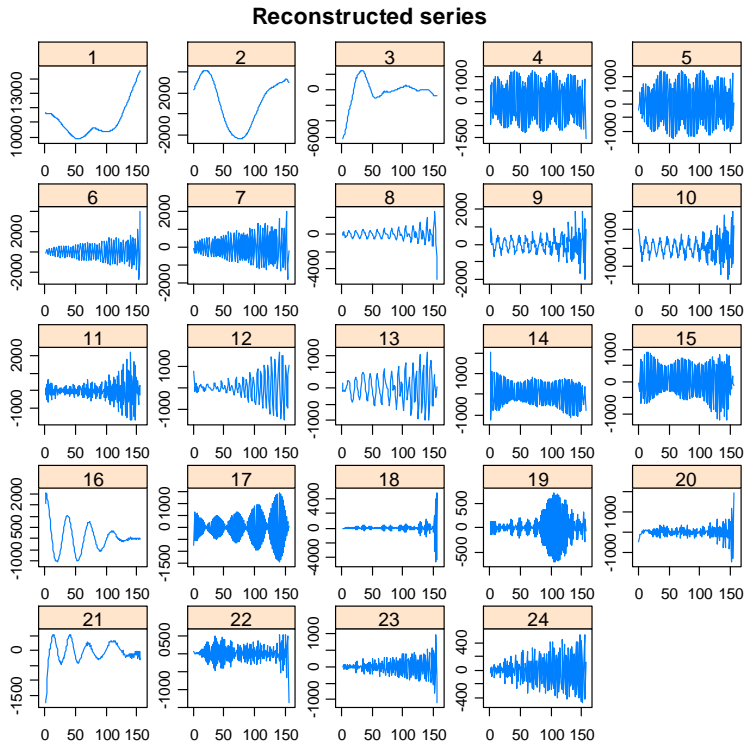
E. Arsitektur DNN Terbaik



Lampiran 59. SSA-DNN *Inflow* Rp 50.000,-
A. Plot Eigen Value



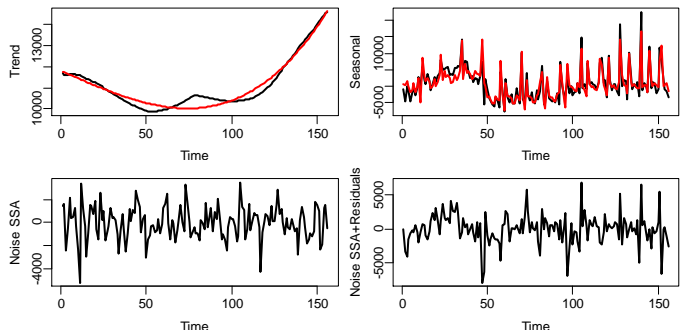
B. Rekonstruksi Individu



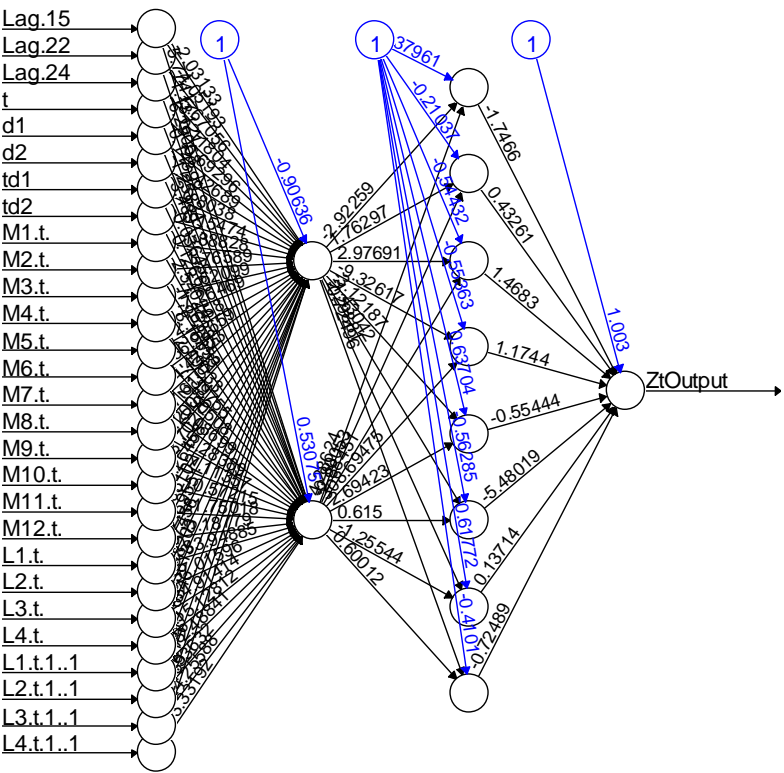
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	1.181e+04	8.341e+01	141.550	< 2e-16	***
t	-3.531e+01	4.586e+00	-7.700	1.62e-12	***
v2	-1.954e-02	6.776e-02	-0.288	0.773	
v3	2.318e-03	2.838e-04	8.168	1.12e-13	***
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	83.376	22.850	3.649	0.000391	***
d1	1888.220	2335.119	0.809	0.420335	
d2	-2372.849	2836.202	-0.837	0.404464	
td1	81.784	30.300	2.699	0.007955	**
td2	-6.257	29.623	-0.211	0.833077	
M1.t.	-1308.424	2542.232	-0.515	0.607727	
M2.t.	-1843.328	2542.747	-0.725	0.469903	
M3.t.	-635.120	2543.262	-0.250	0.803226	
M4.t.	-2368.379	2543.791	-0.931	0.353701	
M5.t.	-4207.722	2544.347	-1.654	0.100791	
M6.t.	-3758.241	2544.943	-1.477	0.142362	
M7.t.	-1618.953	2559.334	-0.633	0.528219	
M8.t.	-4806.715	2569.318	-1.871	0.063807	.
M9.t.	-2692.706	2550.776	-1.056	0.293253	
M10.t.	-257.745	2604.411	-0.099	0.921331	
M11.t.	-5803.849	2776.550	-2.090	0.038701	*
M12.t.	-6528.886	2684.572	-2.432	0.016493	*
L1.t.	13368.342	1813.489	7.372	2.37e-11	***
L2.t.	10516.743	1341.323	7.841	2.05e-12	***
L3.t.	5499.431	1500.210	3.666	0.000369	***
L4.t.	-2353.242	1364.799	-1.724	0.087240	.
L1.t.1..1	2682.047	1796.252	1.493	0.138027	
L2.t.1..1	3643.469	1311.367	2.778	0.006343	**
L3.t.1..1	6986.724	1526.864	4.576	1.16e-05	***
L4.t.1..1	12061.238	1382.750	8.723	1.84e-14	***
tm1.t.	NA	NA	NA	NA	
tm2.t.	-60.143	20.073	-2.996	0.003322	**
tm3.t.	-79.125	20.078	-3.941	0.000137	***
tm4.t.	-61.638	20.087	-3.069	0.002659	**
tm5.t.	-53.448	20.099	-2.659	0.008899	**
tm6.t.	-60.580	20.114	-3.012	0.003167	**
tm7.t.	-59.672	20.895	-2.856	0.005061	**
tm8.t.	-17.128	21.704	-0.789	0.431564	
tm9.t.	-65.161	20.603	-3.163	0.001980	**
tm10.t.	-79.043	20.381	-3.878	0.000172	***
tm11.t.	-39.600	21.864	-1.811	0.072615	.
tm12.t.	-50.337	21.385	-2.354	0.020203	*

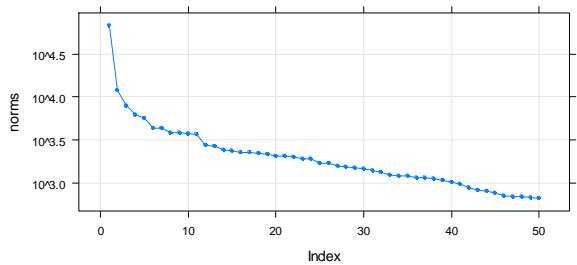
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



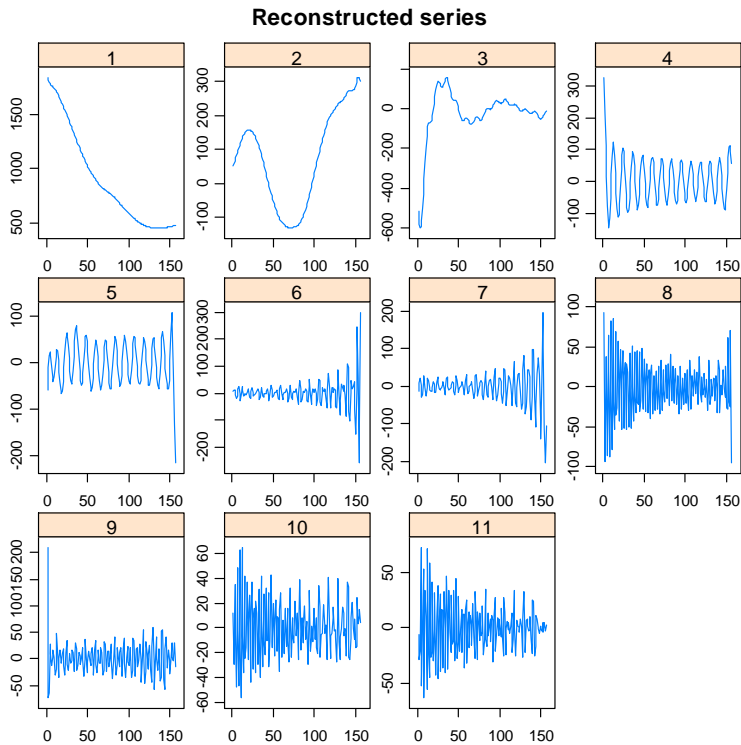
E. Arsitektur DNN Terbaik



Lampiran 60. SSA-DNN Inflow Rp 20.000,-
A. Plot Eigen Value



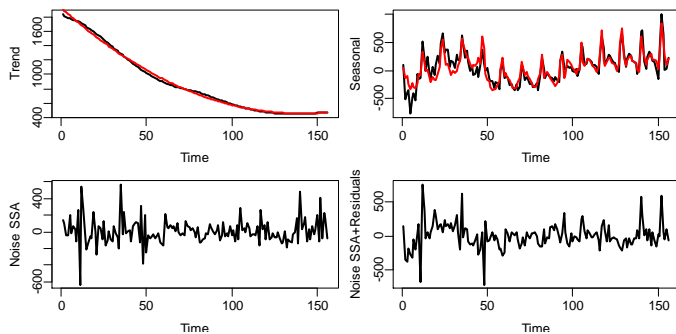
B. Rekonstruksi Individu



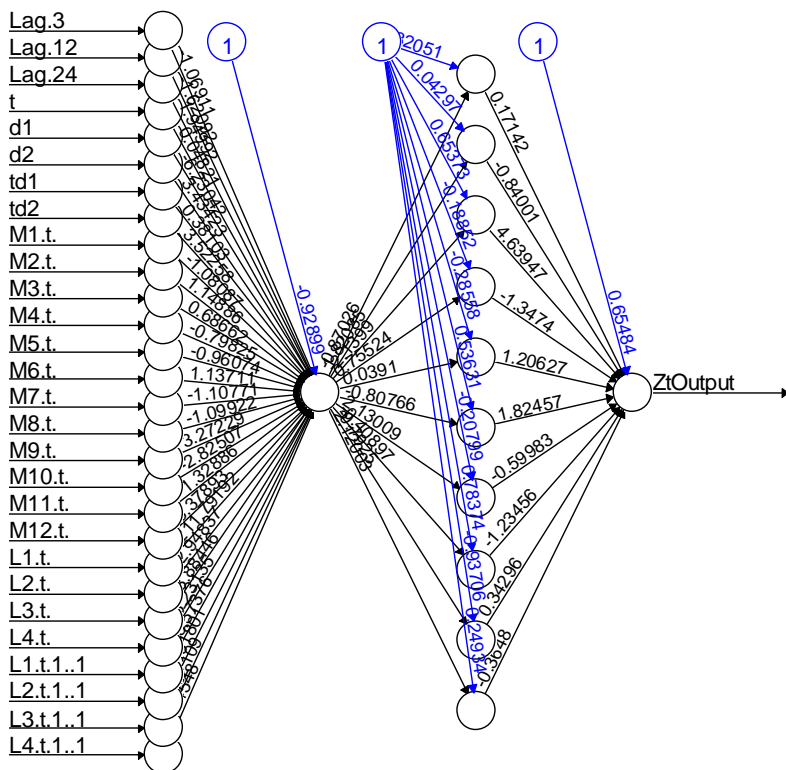
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.931e+03	8.649e+00	223.256	<2e-16 ***
t	-2.144e+01	4.756e-01	-45.088	<2e-16 ***
v2	8.066e-02	7.027e-03	11.479	<2e-16 ***
v3	-2.066e-05	2.943e-05	-0.702	0.484
Regresi <i>dummy</i>				
	Estimate	Std. Error	t value	Pr(> t)
t	1.3579	1.4918	0.910	0.364508
d1	-30.9682	152.4475	-0.203	0.839369
d2	-162.8040	185.1605	-0.879	0.381018
td1	7.2370	1.9781	3.659	0.000378 ***
td2	-0.4744	1.9340	-0.245	0.806650
M1.t.	63.7111	165.9688	0.384	0.701751
M2.t.	-124.7893	166.0024	-0.752	0.453684
M3.t.	-100.5492	166.0360	-0.606	0.545933
M4.t.	-261.6115	166.0706	-1.575	0.117819
M5.t.	-364.5913	166.1069	-2.195	0.030094 *
M6.t.	-244.0308	166.1458	-1.469	0.144510
M7.t.	-345.7427	167.0853	-2.069	0.040669 *
M8.t.	-389.3066	167.7371	-2.321	0.021978 *
M9.t.	-300.0960	166.5266	-1.802	0.074042 .
M10.t.	-123.9160	170.0281	-0.729	0.467546
M11.t.	47.4858	181.2662	0.262	0.793796
M12.t.	25.2271	175.2614	0.144	0.885789
L1.t.	287.4842	118.3931	2.428	0.016657 *
L2.t.	188.2204	87.5678	2.149	0.033608 *
L3.t.	12.8116	97.9407	0.131	0.896144
L4.t.	-178.2340	89.1005	-2.000	0.047718 *
L1.t.1..1	186.8860	117.2678	1.594	0.113641
L2.t.1..1	359.3935	85.6122	4.198	5.2e-05 ***
L3.t.1..1	396.8705	99.6808	3.981	0.000118 ***
L4.t.1..1	235.8549	90.2724	2.613	0.010131 *
tM1.t.	NA	NA	NA	NA
tM2.t.	1.5493	1.3105	1.182	0.239459
tM3.t.	0.6047	1.3108	0.461	0.645403
tM4.t.	1.6827	1.3114	1.283	0.201894
tM5.t.	2.3004	1.3121	1.753	0.082129 .
tM6.t.	0.4782	1.3131	0.364	0.716354
tM7.t.	2.2646	1.3641	1.660	0.099503 .
tM8.t.	4.1268	1.4169	2.912	0.004277 **
tM9.t.	3.9751	1.3450	2.955	0.003760 **
tM10.t.	1.2931	1.3305	0.972	0.333066
tM11.t.	-1.0272	1.4274	-0.720	0.473150
tM12.t.	-0.2145	1.3961	-0.154	0.878165

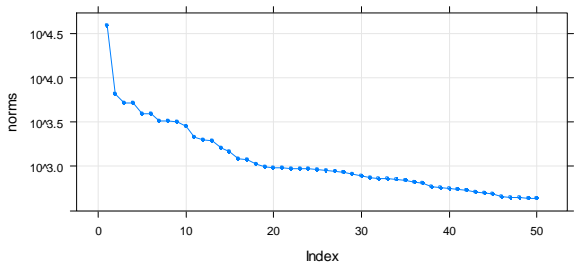
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



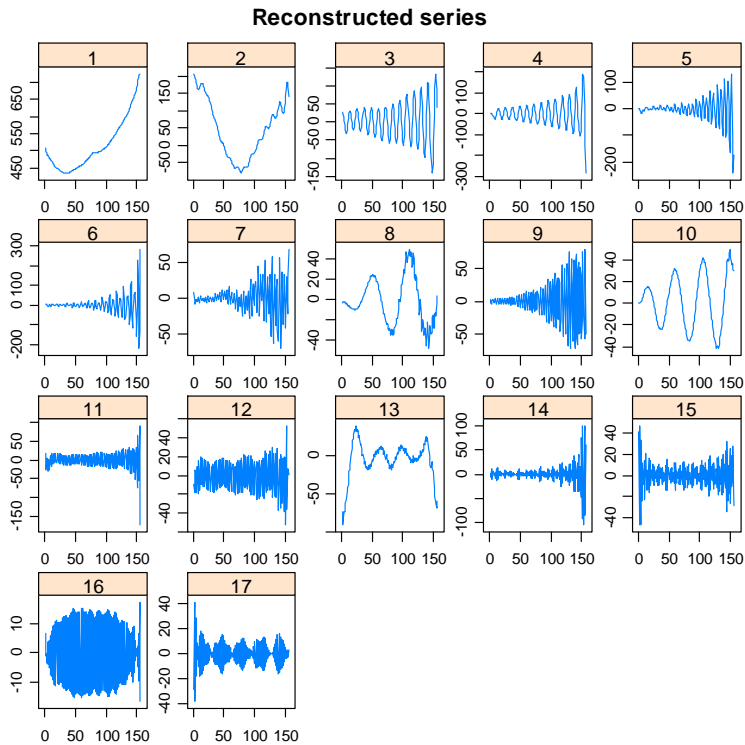
E. Arsitektur DNN Terbaik



Lampiran 61. SSA-DNN *Inflow* Rp 10.000,-
A. Plot Eigen Value



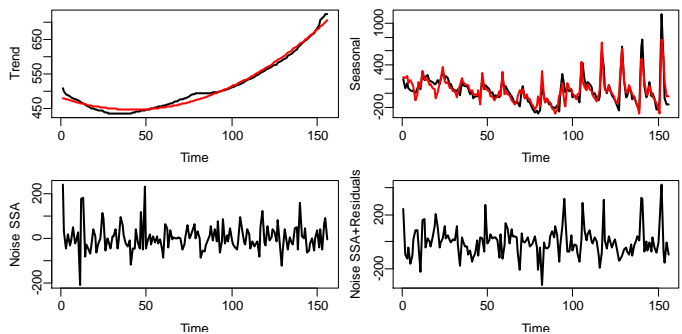
B. Rekonstruksi Individu



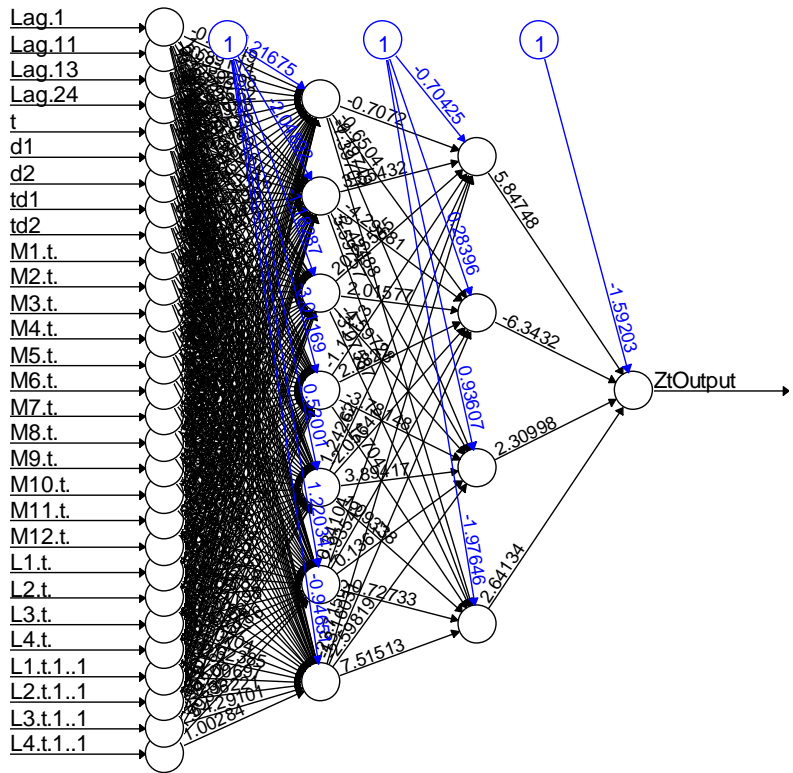
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	4.835e+02	3.228e+00	149.784	< 2e-16	***
t	-1.698e+00	1.775e-01	-9.567	< 2e-16	***
v2	2.040e-02	2.622e-03	7.780	1.03e-12	***
v3	-2.484e-06	1.098e-05	-0.226	0.821	
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	-0.7954	1.0651	-0.747	0.456637	
d1	12.6487	108.8450	0.116	0.907682	
d2	53.4464	132.2016	0.404	0.686727	
td1	-1.7473	1.4123	-1.237	0.218444	
td2	-3.1560	1.3808	-2.286	0.024034	*
M1.t.	217.3208	118.4990	1.834	0.069140	.
M2.t.	217.2982	118.5230	1.833	0.069224	.
M3.t.	242.3701	118.5470	2.045	0.043091	*
M4.t.	152.3853	118.5717	1.285	0.201206	
M5.t.	189.4345	118.5976	1.597	0.112832	
M6.t.	136.8460	118.6254	1.154	0.250957	
M7.t.	97.5968	119.2961	0.818	0.414918	
M8.t.	-120.3995	119.7615	-1.005	0.316762	
M9.t.	-41.2459	118.8972	-0.347	0.729272	
M10.t.	164.0183	121.3973	1.351	0.179209	
M11.t.	366.8016	129.4211	2.834	0.005392	**
M12.t.	228.0924	125.1338	1.823	0.070825	.
L1.t.	-5.3927	84.5307	-0.064	0.949239	
L2.t.	-79.9094	62.5220	-1.278	0.203681	
L3.t.	-231.0415	69.9281	-3.304	0.001257	**
L4.t.	-251.2264	63.6163	-3.949	0.000133	***
L1.t.1..1	46.7002	83.7273	0.558	0.578043	
L2.t.1..1	220.1891	61.1257	3.602	0.000460	***
L3.t.1..1	349.1305	71.1705	4.906	2.96e-06	***
L4.t.1..1	120.8215	64.4530	1.875	0.063284	.
tm1.t.	NA	NA	NA	NA	
tm2.t.	-0.6521	0.9357	-0.697	0.487168	
tm3.t.	-1.1405	0.9359	-1.219	0.225392	
tm4.t.	-0.7932	0.9363	-0.847	0.398557	
tm5.t.	-1.4789	0.9368	-1.579	0.117058	
tm6.t.	-1.4151	0.9375	-1.509	0.133843	
tm7.t.	-0.2182	0.9740	-0.224	0.823148	
tm8.t.	4.3273	1.0117	4.277	3.82e-05	***
tm9.t.	4.3059	0.9603	4.484	1.69e-05	***
tm10.t.	0.8816	0.9500	0.928	0.355259	
tm11.t.	-1.8479	1.0191	-1.813	0.072303	.
tm12.t.	-0.9367	0.9968	-0.940	0.349259	

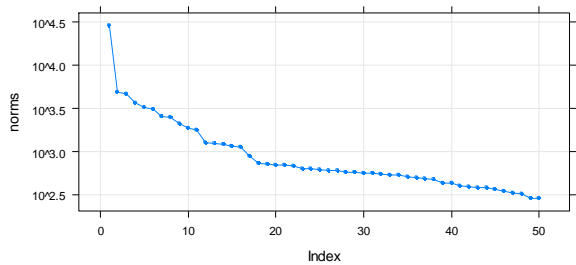
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



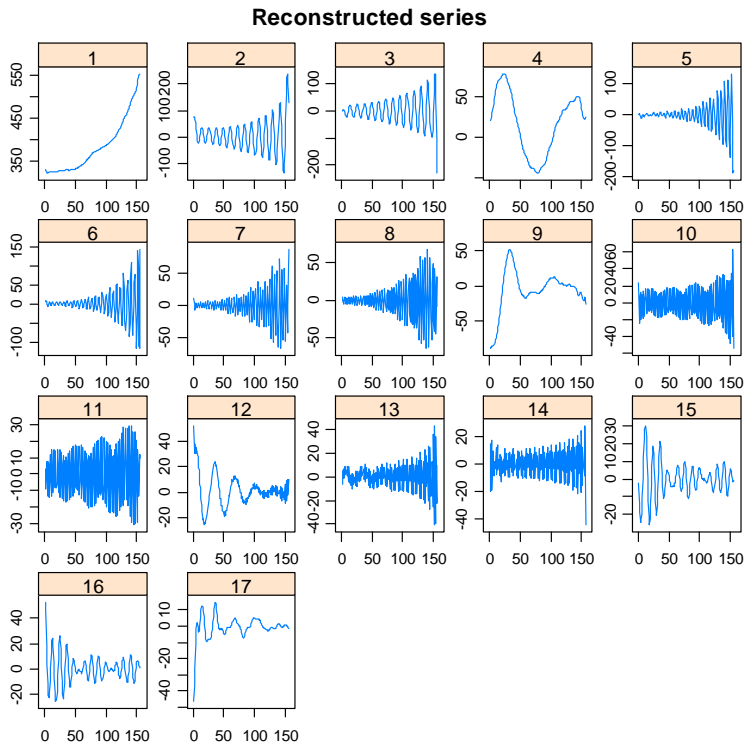
E. Arsitektur DNN Terbaik



Lampiran 62. SSA-DNN *Inflow* Rp 5.000,-
A. Plot Eigen Value



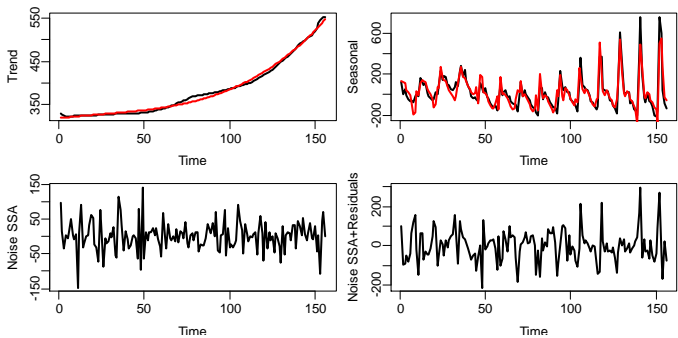
B. Rekonstruksi Individu



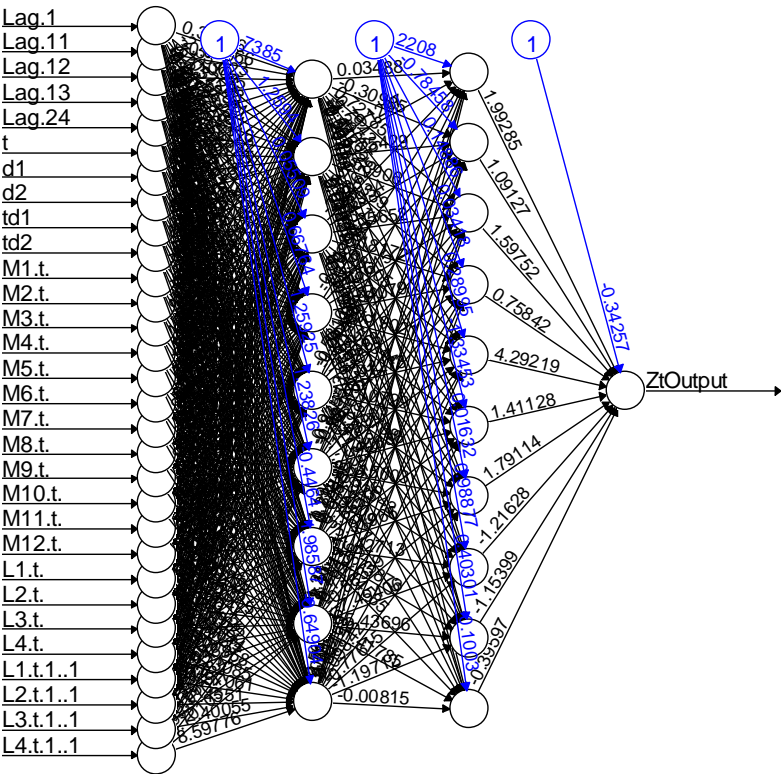
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.184e+02	1.774e+00	179.463	< 2e-16 ***
t	4.778e-01	9.756e-02	4.898	2.46e-06 ***
v2	-6.063e-03	1.441e-03	-4.206	4.42e-05 ***
v3	7.933e-05	6.036e-06	13.142	< 2e-16 ***
Regresi <i>dummy</i>				
	Estimate	Std. Error	t value	Pr(> t)
t	-0.75314	0.83467	-0.902	0.368694
d1	-2.57869	85.29745	-0.030	0.975932
d2	-56.04284	103.60103	-0.541	0.589546
td1	1.79134	1.10679	1.618	0.108181
td2	-0.29423	1.08209	-0.272	0.786158
M1.t.	134.97505	92.86289	1.453	0.148699
M2.t.	122.65232	92.88169	1.321	0.189174
M3.t.	115.40094	92.90052	1.242	0.216585
M4.t.	41.93491	92.91984	0.451	0.652586
M5.t.	25.35870	92.94014	0.273	0.785438
M6.t.	-10.42563	92.96191	-0.112	0.910892
M7.t.	-81.93043	93.48758	-0.876	0.382576
M8.t.	-235.44079	93.85229	-2.509	0.013454 *
M9.t.	-238.30354	93.17498	-2.558	0.011786 *
M10.t.	-4.21062	95.13415	-0.044	0.964771
M11.t.	232.29253	101.42207	2.290	0.023747 *
M12.t.	156.04077	98.06230	1.591	0.114186
L1.t.	-14.98096	66.24331	-0.226	0.821469
L2.t.	-171.35076	48.99597	-3.497	0.000660 ***
L3.t.	-252.22091	54.79981	-4.603	1.04e-05 ***
L4.t.	-242.01758	49.85352	-4.855	3.67e-06 ***
L1.t.1..1	98.48403	65.61367	1.501	0.135990
L2.t.1..1	110.05738	47.90173	2.298	0.023319 *
L3.t.1..1	144.64699	55.77344	2.593	0.010683 *
L4.t.1..1	-46.23040	50.50923	-0.915	0.361877
tM1.t.	NA	NA	NA	NA
tM2.t.	-0.06280	0.73324	-0.086	0.931886
tM3.t.	-0.44321	0.73342	-0.604	0.546784
tM4.t.	-0.02927	0.73373	-0.040	0.968240
tM5.t.	-0.28717	0.73416	-0.391	0.696375
tM6.t.	-0.17488	0.73472	-0.238	0.812265
tM7.t.	1.05959	0.76326	1.388	0.167635
tM8.t.	4.65397	0.79281	5.870	3.96e-08 ***
tM9.t.	5.89550	0.75257	7.834	2.12e-12 ***
tM10.t.	2.52002	0.74447	3.385	0.000962 ***
tM11.t.	-0.93027	0.79865	-1.165	0.246408
tM12.t.	-0.57449	0.78114	-0.735	0.463502

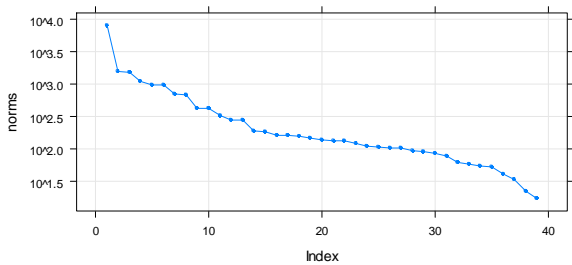
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



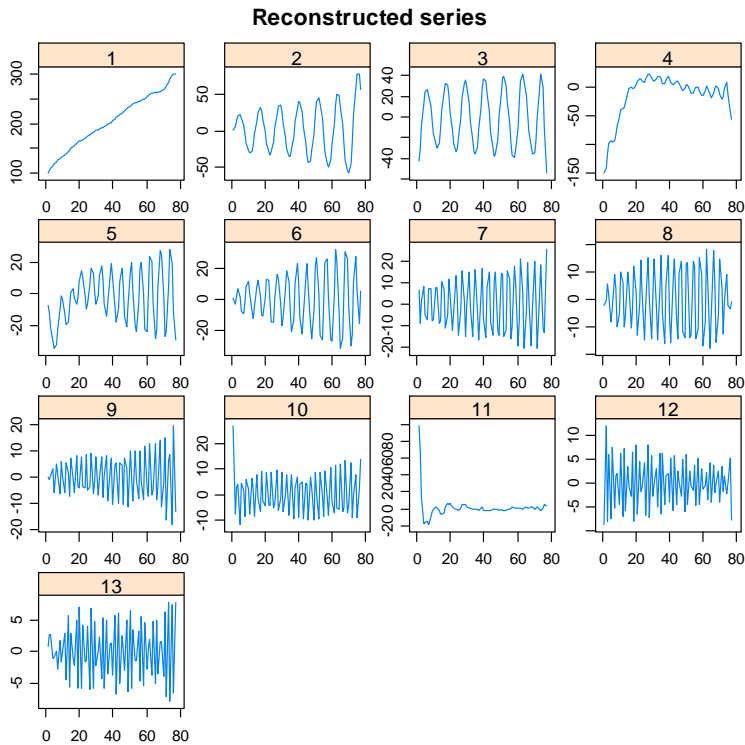
E. Arsitektur DNN Terbaik



Lampiran 63. SSA-DNN *Inflow* Rp 2.000,-
A. Plot Eigen Value



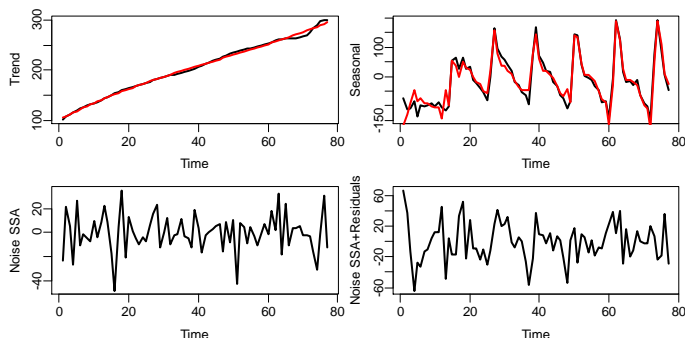
B. Rekonstruksi Individu



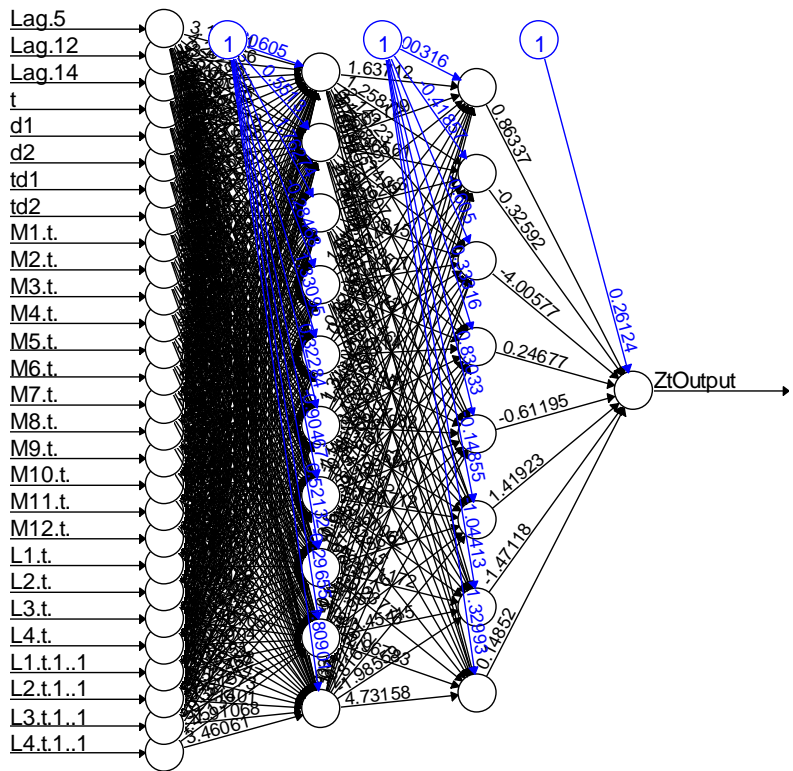
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	1.010e+02	1.661e+00	60.788	< 2e-16	***
t	3.645e+00	1.832e-01	19.893	< 2e-16	***
v2	-3.328e-02	5.439e-03	-6.118	4.27e-08	***
v3	2.424e-04	4.586e-05	5.286	1.25e-06	***
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	-1.299e+00	6.198e-01	-2.096	0.041734	*
d1	NA	NA	NA	NA	
d2	-8.362e+02	1.521e+02	-5.497	1.73e-06	***
td1	NA	NA	NA	NA	
td2	8.167e+00	1.708e+00	4.782	1.90e-05	***
M1.t.	1.775e+02	7.297e+01	2.433	0.019014	*
M2.t.	1.336e+02	7.328e+01	1.823	0.074886	.
M3.t.	1.254e+02	7.367e+01	1.703	0.095557	.
M4.t.	1.363e+02	7.415e+01	1.838	0.072681	.
M5.t.	1.556e+02	7.473e+01	2.083	0.043004	*
M6.t.	1.490e+02	7.541e+01	1.976	0.054348	.
M7.t.	-6.565e+01	9.236e+01	-0.711	0.480903	.
M8.t.	-1.451e+02	7.594e+01	-1.911	0.062394	.
M9.t.	1.673e+02	9.833e+01	1.702	0.095742	.
M10.t.	2.611e+02	8.108e+01	3.220	0.002379	**
M11.t.	2.331e+02	6.285e+01	3.709	0.000570	***
M12.t.	1.754e+02	6.329e+01	2.772	0.008072	**
L1.t.	NA	NA	NA	NA	
L2.t.	-2.022e+02	3.585e+01	-5.640	1.07e-06	***
L3.t.	-1.361e+02	3.224e+01	-4.223	0.000116	***
L4.t.	-1.228e+02	2.844e+01	-4.317	8.57e-05	***
L1.t.1..1	NA	NA	NA	NA	
L2.t.1..1	-4.549e+01	2.843e+01	-1.600	0.116646	.
L3.t.1..1	-1.049e+02	3.133e+01	-3.349	0.001650	**
L4.t.1..1	-1.646e+02	3.544e+01	-4.644	2.97e-05	***
tm1.t.	NA	NA	NA	NA	
tm2.t.	2.543e-01	8.442e-01	0.301	0.764582	.
tm3.t.	2.541e-01	8.449e-01	0.301	0.765019	.
tm4.t.	-6.151e-02	8.461e-01	-0.073	0.942372	.
tm5.t.	-4.107e-01	8.478e-01	-0.484	0.630464	.
tm6.t.	-4.126e-01	8.500e-01	-0.485	0.629764	.
tm7.t.	1.464e+00	9.940e-01	1.472	0.147872	.
tm8.t.	3.311e+00	9.067e-01	3.652	0.000676	***
tm9.t.	1.448e+00	9.124e-01	1.587	0.119518	.
tm10.t.	3.200e-01	8.528e-01	0.375	0.709231	.
tm11.t.	-1.448e-01	7.634e-01	-0.190	0.850412	.
tm12.t.	4.257e-03	7.655e-01	0.006	0.995588	.

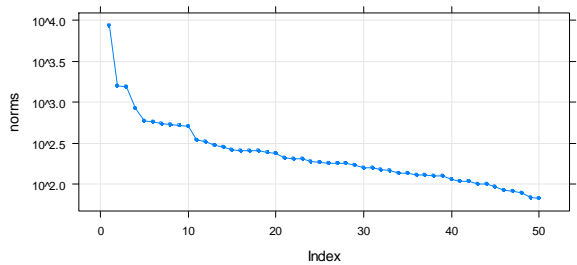
D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



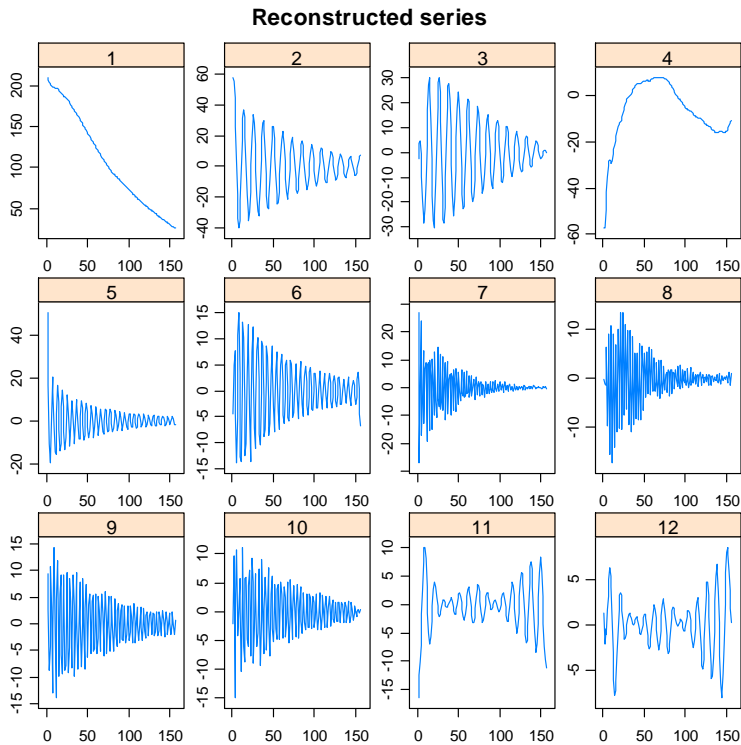
E. Arsitektur DNN Terbaik



Lampiran 64. SSA-DNN *Inflow* Rp 1.000,-
A. Plot Eigen Value



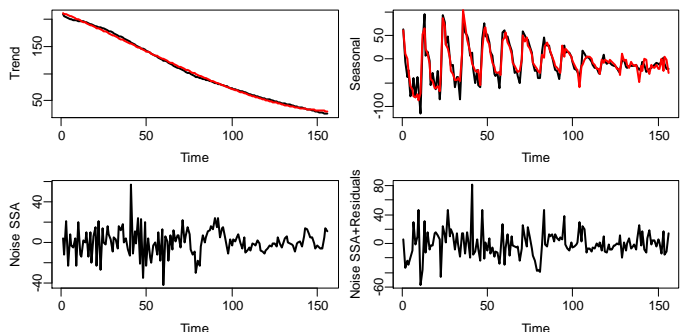
B. Rekonstruksi Individu



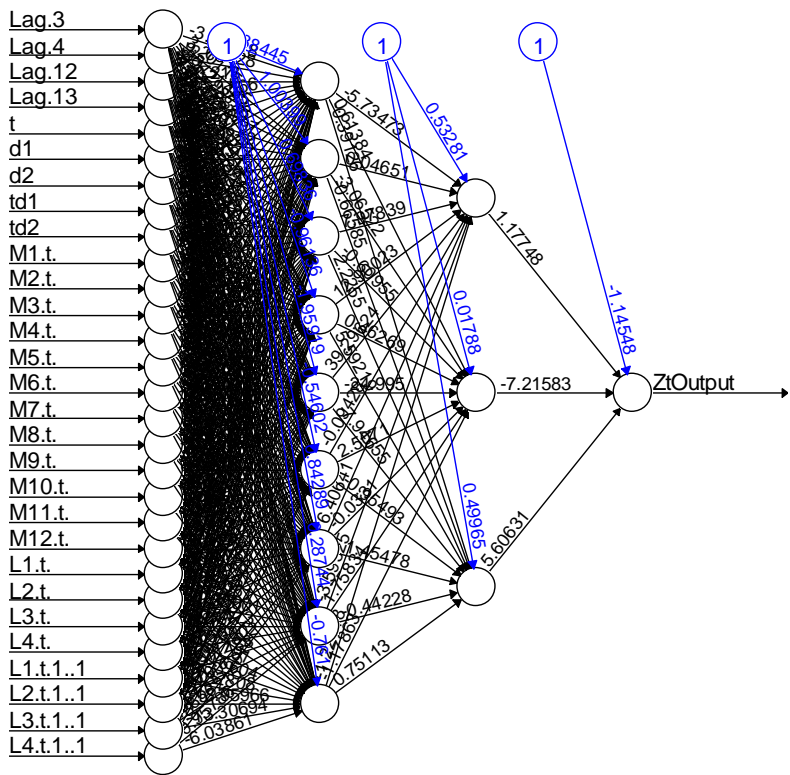
C. Estimasi parameter regresi polinomial & regresi *dummy*

Regresi polinomial					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	2.131e+02	9.171e-01	232.331	< 2e-16	***
t	-1.274e+00	5.042e-02	-25.273	< 2e-16	***
v2	-4.989e-03	7.450e-04	-6.696	3.89e-10	***
v3	3.630e-05	3.120e-06	11.634	< 2e-16	***
Regresi <i>dummy</i>					
	Estimate	Std. Error	t value	Pr(> t)	
t	-1.00400	0.16642	-6.033	1.84e-08	***
d1	-53.59009	17.00750	-3.151	0.002055	**
d2	-1.95779	20.65706	-0.095	0.924651	
td1	1.35854	0.22068	6.156	1.02e-08	***
td2	0.08523	0.21576	0.395	0.693519	
M1.t.	114.24095	18.51597	6.170	9.56e-09	***
M2.t.	62.90510	18.51972	3.397	0.000926	***
M3.t.	51.18272	18.52348	2.763	0.006628	**
M4.t.	27.09489	18.52733	1.462	0.146238	
M5.t.	-10.93835	18.53138	-0.590	0.556126	
M6.t.	-17.03694	18.53572	-0.919	0.359866	
M7.t.	-36.11404	18.64053	-1.937	0.055047	.
M8.t.	-37.60155	18.71325	-2.009	0.046743	*
M9.t.	-33.20643	18.57820	-1.787	0.076399	.
M10.t.	-51.75842	18.96884	-2.729	0.007317	**
M11.t.	-3.42860	20.22259	-0.170	0.865655	
M12.t.	85.91191	19.55269	4.394	2.42e-05	***
L1.t.	20.34483	13.20828	1.540	0.126118	
L2.t.	-7.00501	9.76933	-0.717	0.474742	
L3.t.	-4.45853	10.92656	-0.408	0.683968	
L4.t.	-30.20611	9.94032	-3.039	0.002916	**
L1.t.1..1	48.61611	13.08274	3.716	0.000309	***
L2.t.1..1	38.43795	9.55115	4.024	0.000100	***
L3.t.1..1	24.82266	11.12069	2.232	0.027463	*
L4.t.1..1	12.46837	10.07106	1.238	0.218118	
tM1.t.	NA	NA	NA	NA	
tM2.t.	0.48409	0.14620	3.311	0.001228	**
tM3.t.	0.54553	0.14624	3.730	0.000293	***
tM4.t.	0.70050	0.14630	4.788	4.85e-06	***
tM5.t.	0.99512	0.14638	6.798	4.38e-10	***
tM6.t.	0.97614	0.14650	6.663	8.58e-10	***
tM7.t.	1.13380	0.15219	7.450	1.58e-11	***
tM8.t.	1.08816	0.15808	6.884	2.85e-10	***
tM9.t.	1.05936	0.15006	7.060	1.17e-10	***
tM10.t.	1.37727	0.14844	9.278	9.00e-16	***
tM11.t.	1.01729	0.15924	6.388	3.32e-09	***
tM12.t.	0.26386	0.15575	1.694	0.092835	.

D. Rekonstruksi agregat, hasil *Trend Analysis* & Regresi Dummy



E. Arsitektur DNN Terbaik



Lampiran 65. Weights DNN untuk Peramalan Outflow Rp 100.000,-

[[1]]					
[[1]] [[1]]					
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.25059691	-0.35420446	1.07228506	2.2749677	-2.39297341
[2,]	-2.98641117	-0.45015625	16.13946028	0.1819119	0.60014616
[3,]	1.47205658	2.23894631	2.60670054	-2.0898108	0.61065994
[4,]	0.08057501	-0.04865013	-0.19093542	-1.4726726	0.87369887
[5,]	0.32198203	0.11051725	-4.27344218	1.9551067	-1.28353118
[6,]	-4.25410376	1.67901938	0.46072513	-4.5452869	1.16060550
[7,]	1.03056997	-0.88663734	3.84370687	4.3786437	2.70994937
[8,]	0.11013810	-0.73595531	-2.25332322	0.3810412	1.67714159
[9,]	0.23961415	-0.09889337	1.91635334	0.1406520	-1.37429518
[10,]	-1.01357493	-1.40603137	-2.19111438	-1.8393017	-0.86354207
[11,]	1.81619452	-0.80323766	0.02765222	-0.1729674	-0.02145681
[12,]	0.76570009	0.55541740	5.42797347	-4.0704905	-34.55869846
[13,]	-1.07165050	0.82157234	1.95195298	0.1083318	0.49950566
[14,]	-1.91033360	0.79297891	-0.56781967	1.3882071	-35.46688064
[15,]	0.67452100	0.79317763	1.36781217	0.1312030	-0.12823633
[16,]	-2.32745896	71.48576239	12.92459082	11.8351922	-3.35635684
[17,]	1.46336943	-0.72097085	3.42838331	17.2145732	1.69025898
[18,]	2.24968678	6.02665198	-2.88113721	19.3445430	1.21769265
[19,]	0.66882199	-2.13723767	0.31268012	-1.7390181	-582.15780669
[20,]	1.71953011	-0.31260298	-4.48391868	-48.0986374	-3.81386213
[21,]	-1.15303582	82.20270581	-1.90405985	-8.1849768	-51.44989671
[22,]	-1.57042382	2.30039149	-1.48172292	-1.6214281	2.12262140
[23,]	2.11114994	-1.50416039	10.26692082	0.3028863	-0.94971971
[24,]	-3.82711803	-0.37959772	14.06819123	-0.1003092	0.09232437
[25,]	-0.07380306	1.53767710	12.72772216	0.4433778	3.44905830
[26,]	5.76858304	-2.07361282	-0.12858027	1.5367213	-5.82286406
[27,]	2.98065176	-7.09341855	-9.01080670	-40.9377825	-1.08151256
[28,]	-167.51443619	-374.00265328	-32.81415743	-2.1757808	8.83961808
[29,]	-1.66112684	1.19422722	-1.59094824	-3.3106312	-14.68573594
[30,]	0.41794156	1.51732212	-9.19292700	-0.1304574	2.42266837
[31,]	-0.34605255	-1.71593981	-2.43946155	-5.5997292	3.21739170
[32,]	0.08704710	-1.09528343	3.18859243	29.7334969	-3.28963087
[33,]	34.07474091	0.40811903	15.16736174	-19.4568854	-2.56890632
[34,]	-0.05380504	-1.24340489	10.14256462	-2.2487530	-1.30538483
[35,]	-156.44120139	-710.70945883	-3.26280446	0.3962746	-0.57129609
	[,6]	[,7]	[,8]	[,9]	
[1,]	0.41983974	0.7629771	-1.9759177	0.4919651	
[2,]	0.30517850	-0.2408274	3.1999597	-0.5391844	
[3,]	0.11611322	-2.5850593	-1.8644719	-7.6750369	
[4,]	0.09732042	-1.4167887	-1.0704373	-12.1007767	
[5,]	-0.24932844	-4.0651543	4.9064322	-1.9503725	
[6,]	-0.05891041	0.9940546	0.0123196	-2.2011128	
[7,]	-0.13069087	-2.0922861	-0.2196461	-5.0831317	
[8,]	0.86572791	-1.0800763	5.7328823	-1.0774338	
[9,]	-0.42524880	3.3330598	-3.1773720	-1.3805964	
[10,]	0.19155373	-2.4615865	1.8189986	-5.1166667	
[11,]	-0.01660844	-0.9278967	1.1641317	-0.1284827	
[12,]	1.85539523	5.1749029	-2.0378764	-0.5701020	
[13,]	-22.81126068	1.3445642	-0.1186058	4.2441649	
[14,]	-0.05204064	0.5559327	0.7204874	-0.2951781	
[15,]	-14.88084133	0.8095439	-1.1192751	5.7784293	
[16,]	38.28732649	3.2964897	-2.2028039	-79.9159258	
[17,]	2.81693968	1.7481248	2.3898476	27.7804251	
[18,]	2.72677802	-4.6071193	3.4198134	15.5216607	
[19,]	1.96834750	4.6729760	-4.1724279	-3.8106259	
[20,]	0.74038778	1.7946931	42.1320812	0.9427261	
[21,]	0.15761400	-1.7625964	0.6068041	7.0501511	
[22,]	1.52636119	-3.1964297	2.5852747	3.1263848	
[23,]	1.53683717	-3.1405028	43.6198846	-2.6853317	
[24,]	1.45537425	-2.2818938	5.6804710	42.7455218	
[25,]	3.41671097	0.6077366	0.8459415	-17.1921609	
[26,]	1.85653943	3.5667684	-55.8517776	-3.0126105	
[27,]	0.86924153	1.791046	5.0740410	6.5780333	
[28,]	-3.13318148	-329.9378183	1.4444934	-360.5842448	
[29,]	-3.83436204	3.0424516	-3.0620717	9.2476740	
[30,]	0.13632966	1.9243284	1.4208851	4.7689792	
[31,]	0.56619610	-1.1974214	1.1302073	1.3216518	
[32,]	16.16491856	2.7560673	-4.1967221	3.1874613	
[33,]	-2.75134268	-3.1276782	40.4408840	-21.8167055	
[34,]	-3.38100255	-2.8069658	-4.5975550	24.3678262	

[35,] -2.72320722 -702.0191059 0.9198037 2.8662584

[[1]][[2]]

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1.23375698	0.4327241	0.8325945	-1.6699619	-0.608433
[2,]	-0.38391232	-0.3039460	1.0606132	0.2974572	-1.795875
[3,]	-0.07795626	-2.4008769	1.4101489	4.3689255	2.785372
[4,]	-2.16537763	-1.0813796	-1.8440751	1.7415012	-3.228268
[5,]	-0.11988872	-1.2192288	-1.1051756	1.3986844	6.329186
[6,]	2.93636721	2.2429839	-15.6219177	-3.3231492	-3.393221
[7,]	-0.05689353	-5.4738922	0.3133450	5.5768818	6.096242
[8,]	-0.19176500	-1.3697061	1.7619815	1.2148041	1.940331
[9,]	-1.70294495	1.3431723	-0.3608989	-0.9168877	-1.531618
[10,]	0.11563037	-1.5444199	-1.7725081	7.1016677	2.553838

[[1]][[3]]

	[,1]
[1,]	0.8917556
[2,]	3.2967709
[3,]	5.0340221
[4,]	-0.3507582
[5,]	-1.42221637
[6,]	-0.7665714

Lampiran 66. Weights DNN untuk Peramalan Outflow Rp 50.000,-

[[1]]

[[1]]

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	3.88135302	1.06160741	0.3276573	0.7102393	-1.27848138	-1.318721e-01	-1.28047903
[2,]	-0.65777088	-1.16653518	-0.4054113	3.8776520	9.30365907	-1.485328e-03	14.02620631
[3,]	1.37431894	-2.42606182	-4.3107079	5.6617778	0.18879627	5.952864e+00	10.93425248
[4,]	-0.10429898	-0.37060495	1.4552988	-2.1412832	2.99506206	4.909944e+00	0.92199937
[5,]	-1.66106793	-2.18867201	-0.4597885	0.3286221	3.02659961	9.841972e+00	13.10432834
[6,]	-0.03789696	-0.02051635	-0.8224011	0.6505848	0.09936401	4.239784e-02	0.07346051
[7,]	0.21573956	-1.10823217	-0.2796982	0.1530234	0.26671821	-1.604775e-01	-5.83567620
[8,]	-9.74608912	-16.55319543	2.0537708	5.8687818	-25.45411458	4.626158e+00	-12.27813442
[9,]	-2.62502297	0.12629372	0.7582849	-0.8674860	-1.58563483	-2.467311e-01	-0.03229036
[10,]	0.06825544	7.35537762	0.3850526	-0.3866433	0.47103117	8.909788e+00	-14.35170670
[11,]	59.80832000	38.6872350	16.9051800	-62.5427640	11.33209714	2.621017e+01	-719.68721823
[12,]	3.00553459	38.33074006	-3.3469080	-55.2591138	21.62972192	-1.923073e+00	20.48037872
[13,]	41.27687832	-2.44410776	24.2189028	-9.5995045	48.94182233	4.500262e+01	-0.78934571
[14,]	2.49311397	1.39562189	2.8550317	-3.8254698	12.27014105	-9.431337e+00	31.55900220
[15,]	-0.66886935	38.45160946	6.8653520	-11.7128104	48.62082325	9.826762e+00	10.72885255
[16,]	5.91295807	-0.78586507	6.0569206	-5.0217912	-7.66725233	1.557602e+01	-0.81171913
[17,]	2.53943375	-3.49084468	-3.7709807	-10.6228041	-13.91052247	3.837289e+00	-0.17168912
[18,]	3.13225549	11.55367869	3.7811504	-8.3253937	-4.00010898	3.059460e+00	-1.63842394
[19,]	2.08400219	86.87646183	3.6331267	2.5125748	40.26549411	8.904356e+01	4.19791565
[20,]	7.59344576	-2.14715148	13.6021258	-8.5708439	-6.03555464	1.196951e+01	-28.28046389
[21,]	5.99175269	4.15506344	-1.1253779	-3.4797271	9.49180141	4.882278e+00	-20.77663597
[22,]	-1.94163861	-1.58252979	3.6799575	-0.8556306	-2.12242292	-7.553245e+00	-12.42223641
[23,]	-2.02578898	2.36195637	-5.8627526	4.0023651	-16.45563295	1.784547e+01	1.69778103
[24,]	3.90646978	-2.37588607	-691.6600709	36.2237309	-2.90965516	3.479086e-01	-1.62300877
[25,]	-2.99708870	45.2531186	22.7031320	-60.4904131	35.48138247	3.356710e+01	1.38011815
[26,]	59.35144319	5.60535848	0.8181386	1.6194567	-12.92049174	8.719373e+00	6.13015303
[27,]	32.03684312	-1.63198005	-690.7285032	-114.3813120	36.51530238	3.889204e-03	1.69831593
[28,]	-5.37749721	32.13525879	12.7461767	40.3383822	-8.85334976	-4.979231e+02	27.47723904
[29,]	-3.61642209	-621.61310678	-670.5530001	30.6771587	-2.73345394	-7.552131e+02	6.36933717
[30,]	-4.71747093	-10.23814247	-4.1132325	2.5871825	46.35654750	7.136243e-01	-9.87639414

[[1]][[2]]

	[,1]	[,2]
[1,]	-0.82380050	-0.8141187
[2,]	30.0949085	-1.4594883
[3,]	5.4888032	-1.0860296
[4,]	0.2480887	0.8765858
[5,]	-2.6432862	2.0108358
[6,]	-12.8296597	-0.8207356
[7,]	9.4140108	-0.4569583
[8,]	5.7652895	0.2353314

[[1]][[3]]

	[,1]
[1,]	0.7756612
[2,]	-2.3513139
[3,]	8.5199541

Lampiran 67. *Weights* DNN untuk Peramalan *Outflow* Rp 20.000,-

```

[[1]]
[[1]][1]]
      [,1]
[1,] 1.1116825380
[2,] 0.0785278726
[3,] -0.2371453156
[4,] -0.0798736323
[5,] 0.1404053746
[6,] -0.1567630107
[7,] 0.0411502343
[8,] -0.0026282632
[9,] -1.6466191331
[10,] -0.0446656337
[11,] 0.0279593340
[12,] -0.0005806006
[13,] 0.7295170212
[14,] 1.0895331429
[15,] 1.1511945447
[16,] 1.0563162830
[17,] 1.0527847564
[18,] 1.0160089983
[19,] 0.7898136623
[20,] 0.9726304259
[21,] 1.2155616220
[22,] 1.5711244248
[23,] 1.4136250812
[24,] 0.7387302866
[25,] -0.9223640571
[26,] -0.2937974859
[27,] -0.0628998487
[28,] 0.2658673043
[29,] -0.1928815581
[30,] -0.7275032282
[31,] -1.2126813675
[32,] -1.1173059817

[[1]][[2]]
      [,1]      [,2]
[1,] 1.027342 -8.179097
[2,] -3.166192 13.758995

[[1]][[3]]
      [,1]
[1,] -0.5008512
[2,] 16.2520254
[3,] -2.5811715

```

Lampiran 68. Weights DNN untuk Peramalan Outflow Rp 10.000,-

[[1]]				
[[1]] [[1]]				
	[,1]	[,2]	[,3]	[,4]
[1,]	-0.2647388	-1.26565822	-0.87781840	0.16165245
[2,]	3.7196835	3.85935485	-0.70015410	-2.23997186
[3,]	0.8262371	-4.82542593	-2.28220688	2.45265200
[4,]	3.5233029	2.40596922	5.46987286	1.78032743
[5,]	-2.4675658	-0.74703168	-2.23610637	-3.14948216
[6,]	0.4917730	-1.95626630	1.45055339	-1.65351916
[7,]	-0.7067360	0.97385727	-1.77986058	1.42744042
[8,]	0.5734432	1.88397609	-0.07668753	-3.70080403
[9,]	-3.1924407	-0.47367703	0.05783060	-0.26546941
[10,]	4.8714285	-0.27279650	-0.20819867	1.61021454
[11,]	1.9929666	2.13964463	-2.89729797	0.16230960
[12,]	2.7054322	0.75386678	-0.18996906	0.10342666
[13,]	3.0054323	-0.01351126	2.37608337	0.25158919
[14,]	7.7308605	-2.21109015	-7.21790220	1.87267176
[15,]	6.7413858	-3.19146332	-3.52999313	-5.02942379
[16,]	8.4807760	23.99389750	-11.30438811	-0.21246654
[17,]	7.2106203	-6.49416268	-4.19849398	0.53380745
[18,]	7.5927229	21.76832117	-11.72586192	0.47338687
[19,]	10.0157509	26.88579173	-1.42962309	1.01471481
[20,]	9.5293808	25.58703488	-3.41972219	-709.77510776
[21,]	3.6084623	19.57761471	-9.17291341	0.89299100
[22,]	-2.3970097	-7.17383382	-12.80947394	1.89854573
[23,]	4.8808408	-4.10306100	-10.02372238	5.45995833
[24,]	3.9890085	-10.75508889	-10.74901310	-1.08024988
[25,]	10.6036485	-4.73514887	6.72900266	0.08149055
[26,]	4.2043530	-5.95379409	3.29799587	-4.96797407
[27,]	10.0468356	-690.77814286	-0.48732590	-52.47536803
[28,]	-0.1722162	16.19844573	20.64337607	-3.19388332
[29,]	3.3793207	-1.48938794	5.41211136	-1.41480514
[30,]	0.9390959	26.95164395	9.38284708	1.94857599
[31,]	5.2331670	-130.46907479	-1.50560511	9.50523304
[32,]	3.2980550	23.65287486	18.15487295	1.40244589
[33,]	3.5326089	-6.29238112	11.43295130	-4.71186107
	[,5]	[,6]	[,7]	
[1,]	-2.5189689	0.20359400	-1.71002574	
[2,]	-4.3476987	-0.38763480	0.86759489	
[3,]	-0.2872151	0.91961483	-0.90134857	
[4,]	5.3227051	-0.37973893	-0.01460799	
[5,]	14.9550508	0.74424525	-1.19772040	
[6,]	1.1802823	-1.18318210	-0.07663783	
[7,]	4.9031577	0.27641906	-1.28503056	
[8,]	3.0702038	-0.35621772	-1.54925117	
[9,]	0.2679263	0.98772386	0.07024515	
[10,]	4.5647594	-0.26723365	1.43897663	
[11,]	0.2586825	0.48002729	-0.86874633	
[12,]	-0.7963264	0.76742274	-0.18905241	
[13,]	-0.2974868	0.93944209	0.17040857	
[14,]	-11.8849395	-1.05321259	-6.54764137	
[15,]	17.8831200	1.01361777	1.61829301	
[16,]	-15.9286853	-1.25417324	-7.21343704	
[17,]	-16.2501862	-0.38058363	2.38618238	

[18,]	-15.0987652	0.20908883	-6.20671825
[19,]	-21.0559534	-0.12475618	2.09817231
[20,]	13.8431595	2.18637793	0.21041619
[21,]	-4.6425269	0.08202324	-0.27545907
[22,]	-3.2921425	2.11210205	3.38130606
[23,]	-1.4143749	1.03434270	14.11819346
[24,]	1.0866995	-0.19817321	3.37464975
[25,]	2.9268995	-1.03922958	4.68081248
[26,]	-2.3588809	-0.13366410	-14.68853157
[27,]	5.2145839	-0.58203619	1.26222637
[28,]	-3.1284026	-0.90301123	-6.13819261
[29,]	5.4669325	-0.80183175	-3.81588772
[30,]	3.6183696	1.32082067	0.23830604
[31,]	-3.0252622	1.03917965	0.14176722
[32,]	1.8142191	0.55811104	-0.48307436
[33,]	-0.9383200	-1.32868715	-8.67546757

[[1]][[2]]

	[,1]	[,2]	[,3]	[,4]
[1,]	0.9521908	1.05956482	0.6619553	0.044430554
[2,]	-0.3930952	1.65299321	-5.0784210	-0.842309734
[3,]	-1.0142959	-0.01096273	1.2662742	-0.720688290
[4,]	1.3511306	7.03448945	8.1518953	-0.004169606
[5,]	-2.7934314	-6.76010681	-0.5165721	3.234631207
[6,]	2.0605668	-78.02216228	-10.1208618	67.586239151
[7,]	-1.0315837	0.51705934	0.4174779	-0.210158358
[8,]	-3.5562432	3.18118853	4.6062953	-1.367238722

	[,5]	[,6]	[,7]
[1,]	-1.9488480	1.1677978	0.2607070
[2,]	-4038.9719220	1.7377160	-4.1081453
[3,]	-14.6655070	-1.8294792	1.4431414
[4,]	3.7600514	1.5841963	-0.7720150
[5,]	-0.1115245	-1.3011310	0.6864477
[6,]	1.3131782	-1.1074904	-1.9768852
[7,]	-1.0904116	-0.1472586	0.1615165
[8,]	-3.9464817	-2.7090214	0.9953894

[[1]][[3]]

	[,1]
[1,]	-0.1365352
[2,]	1.8864897
[3,]	1.6192009
[4,]	-1.4228496
[5,]	-0.5179068
[6,]	5.4458042
[7,]	4.4476250
[8,]	-1.7503484

Lampiran 69. Weights DNN untuk Peramalan *Outflow* Rp 5.000,-

[[1]]	[[1]][[1]]	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.37001550	1.4185182	-0.06742541	-0.75595218	0.2679168	

[2,]	-0.71970117	1.0294500	-0.14413367	-0.69007190	0.1441213
[3,]	-0.23781519	0.5858224	-0.89581892	-0.16372877	-0.4682927
[4,]	-0.51680776	-1.1919226	0.28374909	-0.66233616	-0.1701382
[5,]	-0.08172516	0.5020277	-2.30885418	-1.83689768	1.4339761
[6,]	0.28871311	1.6915378	-0.23718697	-2.93340007	-4.0146526
[7,]	1.63298771	0.2322768	-0.14853085	-0.77086657	3.6354133
[8,]	-0.94073929	1.8251611	3.35179733	-0.19848168	-1.7961614
[9,]	0.36730068	-0.5888691	-2.37545451	0.12310283	-1.6225438
[10,]	-0.87828500	-0.0970888	0.13216946	0.07471511	-0.8077313
[11,]	-1.45040155	1.7908847	-0.98032358	2.05383555	-25.0550396
[12,]	2.81296122	-1.9810430	1.48878038	-0.88803866	1.8948051
[13,]	0.97583014	-0.1481552	-0.25726781	2.95862838	-26.6831280
[14,]	0.86226977	0.0417808	-0.22798177	1.58697554	0.7525111
[15,]	-0.86440023	-3.3180591	-0.94038982	26.53666327	0.8763951
[16,]	-0.78146752	0.3172132	1.90136958	-10.21915430	-1.2408542
[17,]	-1.03537486	-406.6000562	-11.76779384	-8.95717596	0.1910861
[18,]	-1.39209537	1.7218210	3.30548547	-6.80661265	1.2812053
[19,]	-1.14693918	-2.6237234	-8.67722721	1.54489936	1.1763148
[20,]	-0.59317414	0.1484507	-11.08271420	1.41839779	1.3821676
[21,]	-0.75830612	0.8688883	-2.30706943	-0.34812295	0.4360997
[22,]	-1.20748181	0.7799320	-2.84430552	0.60770908	-0.5725506
[23,]	-2.31020431	-2.2274554	0.21469472	32.80970556	-386.5292364
[24,]	1.171690086	-0.6808071	3.64396348	26.91735162	2.3231286
[25,]	0.16669515	-0.3312021	3.05449809	25.79316726	1.5460339
[26,]	-2.07053565	2.1908934	5.21611355	-9.53473108	-0.0106779
[27,]	-5.00778488	4.1034004	2.50703562	-93.44539731	-481.5005362
[28,]	0.10032844	0.9118863	1.56230623	2.60524372	7.4501495
[29,]	-0.97443039	-1.0558411	-9.91333679	0.62182549	-0.5733843
[30,]	-0.04200910	-1.3764336	1.87940404	36.58664140	30.1750188
[31,]	1.66811656	0.9655009	-2.01511138	-1.54029502	2.5475513
[32,]	-4.05817679	-64.1814484	-3.26857147	1.68767138	39.6169128
[33,]	0.93726827	2.5902501	-1.02934037	-0.65043796	-3.4776933
[34,]	-4.44915559	1.1564199	2.87269873	-127.90232427	-377.4288799
[, [, [, [, [, [,					
[1,]	-0.98318210	-0.4841788	-0.93170885	-2.550798e+00	0.25320781
[2,]	0.07641906	-0.5841639	-0.19534737	2.047335e+00	-4.44363537
[3,]	-0.55621772	-0.2126531	-3.95259889	1.162921e+01	2.05400948
[4,]	0.78772386	-0.6021921	-0.02339036	5.113487e+00	1.74212845
[5,]	-0.46723365	2.5416418	2.07471080	3.813313e+00	1.69633240
[6,]	0.28002729	-3.0002326	0.34265959	-7.948612e+00	1.69438546
[7,]	0.56742274	2.4333605	1.07189080	-3.454551e+00	1.17904414
[8,]	0.73944209	1.8673560	0.60863547	-5.193087e+00	1.28910784
[9,]	-1.25321259	-2.4380802	0.60719211	-1.134764e+01	2.85465922
[10,]	1.21361777	-1.1978148	0.01020487	5.936086e-03	-0.06190311
[11,]	-1.05417324	6.5487160	-1.31928105	-2.562764e+01	-3.79405448
[12,]	-0.38058363	0.5676027	0.76651636	-1.554201e+01	-1.55481252
[13,]	0.40908883	0.7401097	0.43107173	-2.745211e+01	0.04594873
[14,]	-0.12475618	1.0454549	-0.01268887	-1.437576e+01	0.46639007
[15,]	2.38637793	-2.0656819	-7.76135911	-2.134754e+00	7.66603137
[16,]	0.28202324	3.9224624	1.35997503	-7.966114e+00	-17.42045921
[17,]	2.21210205	4.6291369	-1.21449568	-4.287888e+00	-17.92179052
[18,]	1.13434270	5.0084190	-9.08793298	3.677859e+01	2.29698193
[19,]	-0.09817321	4.3810176	-4.01588786	-4.561007e-01	3.82081684
[20,]	-0.93922958	5.0399286	-2.62227699	4.964183e+00	-5.16530947
[21,]	-0.03366410	4.4345534	-3.01205888	-1.147909e+01	0.82019624
[22,]	-0.48203619	2.4989278	-4.22317228	-6.808191e+02	5.30511560
[23,]	-0.90301123	-1.1640964	-2.35214283	-1.176653e+01	0.68670773
[24,]	-0.80183175	10.2606996	-1.86806803	-6.954522e+00	6.34381164
[25,]	1.32082067	2.6256032	-2.09486050	-4.994312e+00	-16.73785989
[26,]	1.03917965	-368.2216269	-0.42932834	6.227772e+01	1.34079168
[27,]	0.55811104	1.8922942	-0.09775659	1.124282e+02	43.53017692
[28,]	-1.32868715	5.6810464	2.17819483	1.249587e+01	14.87045938
[29,]	-1.92455679	1.4566644	0.61907327	-3.249675e+02	4.42453607
[30,]	0.02269541	1.0406390	5.02205563	-3.853760e-01	1.51667593
[31,]	-0.58148841	-6.7949752	-0.11385411	-7.746425e+00	-6.59950670
[32,]	-0.22415163	-52.3793149	-4.45489355	-4.026097e+00	-3.64348246
[33,]	1.19843419	5.4832761	-0.89790443	-2.242045e+00	1.09679942
[34,]	-1.35857871	-0.4331800	-12.77328764	-4.994622e+02	35.16721777
[[1]] [[2]]					
[, [, [,					
[1,]	0.7682795	-0.9543097			
[2,]	-0.6144740	-1.2707412			
[3,]	-0.7108508	5.8194476			

[4,]	-1.4723111	-0.8649008
[5,]	2.8379926	1.1958304
[6,]	1.6724459	1.1403577
[7,]	2.5479267	0.3728346
[8,]	1.1931436	-1.9469677
[9,]	-2.1395338	0.3472941
[10,]	-0.3647759	0.4517384
[11,]	-2.1661707	-0.4465872
[[1]][[3]]		
	[,1]	
[1,]	1.883229	
[2,]	-4.629416	
[3,]	4.700087	

Lampiran 70. Weights DNN untuk Peramalan *Outflow* Rp 2.000,-

[[1]]				
[[1]][[1]]				
	[,1]	[,2]	[,3]	[,4]
[1,]	2.287247161	0.79490960	1.725357e+00	-0.07933306
[2,]	-1.196771682	-2.60362840	8.411250e-01	-1.17436101
[3,]	-0.694292510	-0.14808131	4.385872e-01	0.30872212
[4,]	-0.412292951	-3.73567102	-1.298079e+00	-1.60387854
[5,]	-0.970673341	0.22848273	9.321255e-02	0.99128963
[6,]	-0.947279945	-0.02620268	1.174696e+00	1.02322044
[7,]	0.748139340	1.45630682	-2.285000e+00	0.84014544
[8,]	-0.116955226	0.28110545	1.282935e+00	0.12007861
[9,]	0.152657626	-1.34640133	-3.667871e-01	-0.42625506
[10,]	2.189978107	-0.10333152	-8.736308e-02	0.45892624
[11,]	0.356986230	-0.39101243	-6.240231e-02	0.64504795
[12,]	2.716751783	-0.40152661	2.422693e+00	0.61153055
[13,]	2.281451926	1.35051758	3.425854e-01	-0.88921129
[14,]	0.324020540	0.59119003	4.248236e-03	1.54389235
[15,]	1.896067067	-55.98850730	7.849155e+00	-1.24176360
[16,]	0.467680511	7.31591651	3.539079e+00	1.10344734
[17,]	-0.893800723	7.69282932	-9.597957e+01	0.98277236
[18,]	-0.307328300	9.37116047	2.382742e+00	0.30432717
[19,]	-0.004822422	-92.75046170	5.958811e+00	-1.55471822
[20,]	0.988164149	4.63402567	-1.906640e-01	1.56989078
[21,]	0.839750360	1.55950812	-3.034036e-01	0.68844969
[22,]	0.705341831	2.07711233	2.110216e+00	-0.17760366
[23,]	1.305964721	-23.26499910	6.290774e+00	0.72920127
[24,]	-1.387996217	4.46091308	-1.933059e+02	1.53325093
[25,]	1.272916864	4.89164242	6.294893e+00	0.50657845
[26,]	0.184192771	-132.48345821	-2.540963e+02	0.03332768
[27,]	0.752279896	-4.99520073	-1.251009e+00	-1.46755072
[28,]	0.591745052	-4.04446161	-9.213398e-01	1.01915772
[29,]	-0.983052596	-0.06057843	9.698782e-01	-0.59333933
[30,]	-0.276063955	1.55191393	4.845337e+00	0.81240989
[31,]	-0.870851023	-0.87536985	-6.299940e+00	0.86616783
[32,]	0.718710553	-1.44111412	-6.336914e+00	0.36830830
[33,]	0.110652878	1.49337868	1.189334e+00	1.13481910
[34,]	-0.078466768	1.74947266	2.821971e+00	-0.75709928
	[,5]	[,6]	[,7]	[,8]
[1,]	0.47082776	-1.80332955	1.72438096	-0.024182912
[2,]	0.44855056	0.06553770	-1.77188283	-0.047280375
[3,]	1.08803225	0.09426554	-1.23344800	-0.142329879
[4,]	0.25283403	-1.85112201	-0.27218170	-0.501145773
[5,]	0.92941696	0.47292365	-0.55070399	8.765873983
[6,]	-10.99375742	1.34665245	3.03076588	2.214966434
[7,]	0.95881922	-0.48671896	-1.39447358	-0.509048228
[8,]	0.21264396	2.70962469	-0.35560761	7.834043852
[9,]	1.45589819	0.89632541	-1.31074127	0.371222465
[10,]	-0.02765267	-0.06256192	-0.06842130	-0.009092716

[11,]	-0.88892843	2.19252993	-0.90189298	-1.828859263
[12,]	0.48594357	2.32711280	1.49159937	0.805924242
[13,]	0.34049676	1.13486960	-0.13727932	-1.477591780
[14,]	-1.09792419	-0.19088456	0.10828170	-2.973255713
[15,]	3.28152217	-38.80235160	3.39519565	2.504381328
[16,]	1.42177050	5.76917469	2.63841068	2.548238819
[17,]	-2.98208182	8.27177031	-4.81490004	4.096898142
[18,]	-0.47224093	3.13388343	5.88219686	0.921202937
[19,]	2.50822592	6.46679978	-121.53985928	1.430150201
[20,]	2.23237377	-2.51730056	2.85136998	-14.367380050
[21,]	0.74230767	-1.76249960	0.52491961	-1.818401228
[22,]	-141.03228060	-0.56279440	0.01386982	-1.605052652
[23,]	-2.14749730	4.71220956	0.78703824	5.364223197
[24,]	-98.10451558	-245.61301240	1.48329724	7.412622558
[25,]	-0.20920363	-242.44914756	1.97072550	1.369735338
[26,]	-4.66949837	-263.40540682	2.21100657	6.122432751
[27,]	1.40025227	4.18926964	6.04354714	17.617992427
[28,]	1.99538217	4.14696910	3.68966204	3.691203089
[29,]	1.88261043	0.37542841	1.30397663	0.395813511
[30,]	-3.58299617	-6.99065986	2.25475560	10.626697932
[31,]	1.51698239	3.21768305	2.71669739	0.292440092
[32,]	3.23541689	3.41285641	1.07699416	3.889947120
[33,]	-0.53144223	-0.28230488	-0.26043105	0.396684652
[34,]	7.31285921	4.38879526	0.30786742	6.910206217

[[1]][[2]]

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.3234520	0.4566725	-0.9173094	-0.8262663	-1.1536277
[2,]	-1.2202654	-1.7288221	1.3923533	0.3483723	0.9312899
[3,]	-1.3432468	-1.1193255	-2.9578184	-2.0717347	-0.5071791
[4,]	0.6017224	-1.5208895	-0.1065216	-2.9773661	-3.6085485
[5,]	0.7906210	-0.7653643	-2.1113252	-0.5651389	1.5963785
[6,]	0.6696241	0.1629529	0.1777022	-0.5790128	-2.9511196
[7,]	2.1057617	2.7772595	-15.2258354	0.6115491	1.9608081
[8,]	1.6656232	0.7678703	-0.6824025	4.4598151	1.7626407
[9,]	1.0582446	1.3543685	-1.5768489	-1.1557946	-0.2616937

	[,6]	[,7]	[,8]
[1,]	-1.2188987	0.2972823	-0.7724977
[2,]	-0.3855674	-0.4469281	-1.0067708
[3,]	-2.5935292	0.7687096	2.2428398
[4,]	-2.5236930	0.5042298	1.9977936
[5,]	-0.6862943	0.8014522	1.2853392
[6,]	1.8284408	-0.2333497	-0.9249774
[7,]	2.2625600	-2.7662913	-1.7622113
[8,]	2.2127296	-1.4197252	-2.0487220
[9,]	0.2478588	-1.7984839	-1.0039696

[[1]][[3]]

	[,1]
[1,]	-0.90838229
[2,]	1.25257016
[3,]	2.33008966
[4,]	-0.58494084
[5,]	0.06632899
[6,]	0.48605710
[7,]	0.97207900
[8,]	-0.23356576
[9,]	-1.61162474

Lampiran 71. Weights DNN untuk Peramalan Outflow Rp 1.000,-

[[1]]					
[[1]] [[1]]					
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.34015593	-0.48566050	-0.35761843	2.54804030	-2.35817979
[2,]	0.53306633	-1.61597978	1.58980339	1.14992008	-12.62987006
[3,]	1.32786226	-0.79258448	1.95334089	2.21868712	-6.99709383
[4,]	1.66014574	-0.44906191	2.19988091	0.07862643	-0.01278975
[5,]	0.12704093	-0.13651257	-0.09033007	0.14916558	3.24884013
[6,]	0.34335269	-0.27358623	0.99300157	-0.52738060	5.05899075
[7,]	0.94032092	-1.84910318	1.89442025	0.24948087	2.25013943
[8,]	-0.06936890	-0.73170506	1.44091784	0.97185947	-6.00936850
[9,]	1.65449776	2.01354747	1.74218386	1.85893982	-7.90705340
[10,]	-0.05704222	0.73249393	-0.43901954	1.06416860	16.03797744
[11,]	0.90965041	2.10092044	1.55727767	0.95598009	0.01509267
[12,]	0.98465418	-0.10548372	-1.92603624	1.53095366	-1.62278103
[13,]	-0.39269536	0.01915576	0.03028513	1.64713685	1.12219178
[14,]	-1.04640172	0.01583855	-1.58997285	0.40659804	-1.75448940
[15,]	1.78222896	-1.08876777	-1.64679282	-0.28331666	1.67729369
[16,]	-2.29870212	-0.53828715	-0.22482961	1.74119983	1.89844223
[17,]	38.07265469	2.16630136	0.92178697	0.63885842	5.48595848
[18,]	1.08198511	-0.46224705	-2.25923714	2.21063159	-0.08187318
[19,]	2.01992313	1.37571551	4.08997572	0.06935903	-2.43858837
[20,]	2.54397169	-0.94757263	1.79949871	-0.54422958	-2.15530615
[21,]	1.46811338	-1.86587824	3.14214037	1.03757952	-0.11260384
[22,]	0.69948948	-0.22297109	0.95263707	-0.16958599	12.23490321
[23,]	0.74570391	1.03586253	-0.01937410	1.13880561	3.58167561
[24,]	-1.68403955	1.23922980	0.52785210	-1.32398983	5.70814709
[25,]	-12.09577692	1.77161877	-1.12146724	-1.34243008	59.49679645
[26,]	-0.57421946	-1.67824221	1.70928090	0.88980417	-3.04792581
[27,]	-1.53493343	2.13124252	5.21893821	0.73096318	-2.33943662
[28,]	-28.83047049	-0.70314433	13.69342140	1.27106771	-0.96599140
[29,]	0.85930327	0.25816476	-4.12055916	-1.61624558	-0.81678496
[30,]	-0.32876329	0.50623480	-5.25310264	1.41257910	-3.97114174
[31,]	1.23129720	-0.81999511	0.97169093	1.30049470	4.63491206
[32,]	-0.92072348	-1.89846699	-9.11336586	1.43206833	22.58146644
[33,]	1.39906195	-0.37929259	-5.79229743	1.18641026	3.21009412
[34,]	-1.58415772	0.08417990	1.90272164	2.51007351	24.20915379
[35,]	-8.16983133	-0.89548661	5.33095773	1.35380985	-7.39242441
	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	-0.62535657	0.92406703	-1.027168783	1.66451504	0.10839256
[2,]	0.48841566	6.50391893	-1.339527126	-0.03695033	-0.11920310
[3,]	-0.18515216	2.95854073	0.140206527	-0.45526592	-0.25447573
[4,]	0.61399731	-3.89555778	-3.751167761	-0.27574965	0.35745479
[5,]	-0.95104759	-0.48509985	-2.067954602	0.64402572	-0.44887752
[6,]	-1.15310059	-0.06308434	-1.091451027	1.06302314	-1.44098463
[7,]	-0.15792100	0.54176113	-2.078263715	-1.07477941	0.96320219
[8,]	0.73177693	-2.25179134	-0.354811085	-0.24333882	0.75417982
[9,]	-1.30154742	-1.09077465	-1.501689079	0.99648595	0.74702846
[10,]	0.64342722	1.48574728	3.194049046	-0.08164261	-0.93227941
[11,]	1.01574354	0.16385675	0.009319758	-1.44566358	0.01955014
[12,]	0.05205584	0.71551914	-0.991292546	5.23151280	0.96878980
[13,]	0.51513317	-0.89536698	-14.778314072	-0.07858007	2.12489356
[14,]	-0.81497516	1.92385165	0.105380029	1.05358941	1.50279345
[15,]	0.50364199	-0.16345156	-14.088071204	1.36534620	-0.09707422
[16,]	-4.72215567	-0.42117006	-5.390458311	1.30107164	1.21734649
[17,]	-2.67782533	0.34201326	-5.060821894	0.25040671	2.00807799
[18,]	-3.25157582	-0.92497731	1.177309739	1.09392123	0.71635681
[19,]	-2.00133690	-0.99983830	1.507214860	-0.20348600	2.25615987
[20,]	32.62737103	2.98772594	-3.187239562	-0.70930248	2.87976493
[21,]	-1.88859545	-1.91662241	-0.214402925	-706.59154973	2.54591280
[22,]	-2.04992722	-2.25752635	1.450773740	-707.93773975	0.97201242
[23,]	-2.27435092	-10.35251683	0.476436215	-0.50247904	1.20243697
[24,]	-1.24826943	0.79527182	5.123444285	-1.54753275	4.03808384
[25,]	-9.76667861	-6.25722163	-3.712952578	-0.54845445	3.34865672
[26,]	17.62139005	-1.76566262	5.657685018	2.16686855	0.65639737
[27,]	-4.46240119	3.65283570	3.201670339	1.43583305	-0.18387417
[28,]	-26.66825862	0.36383038	3.371460878	1.25226577	2.30234845
[29,]	0.21871001	-0.23200935	1.891173542	1.62014724	2.33407096
[30,]	-0.96655433	-6.96026825	-1.584328275	-3.85155910	-0.65157717
[31,]	-0.91339334	25.75387602	-4.002996366	-0.42106022	-0.75438627
[32,]	-1.56157995	12.64706223	-1.419277045	-1.01458027	6.37809066
[33,]	-0.35054757	-11.61689891	21.611503724	1.48715544	4.59308879

[34,]	0.59946362	0.99937924	31.801152173	-2.32164014	-0.20151771
[35,]	-7.16188198	33.84991689	-5.736269934	5.03618728	-4.34977256
[[1]] [[2]]					
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.50345724	0.96840600	-1.3276560	-0.99445337	-0.68889354
[2,]	0.16018805	-0.01219517	0.7121710	1.13269270	0.56437624
[3,]	-1.32220977	-0.95259530	-0.5584901	-1.85369880	-0.55793805
[4,]	0.63223844	-2.04036353	-0.3077304	0.37685888	11.06556521
[5,]	1.37706150	-1.59447837	-0.7736457	-0.30797531	-0.34263686
[6,]	-0.04348121	-0.12191792	2.0141707	1.22937850	0.70879566
[7,]	-1.46250227	2.52579131	-0.3603436	0.64294858	-0.01998232
[8,]	0.30331621	-0.79742975	1.0020561	-0.07719628	-8.83025591
[9,]	0.42751850	0.05684779	-0.1650454	0.68260729	-14.20240591
[10,]	-0.94212766	-0.63693340	-1.4793789	-1.76940953	24.84998591
[11,]	1.36248454	2.12382994	0.6723755	0.91545899	26.05482273
[[1]] [[3]]					
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.2679707	3.8504288	0.39270549	-0.99445337	-0.68889354
[2,]	-0.1733057	-2.4332712	-1.44672511	1.13269270	0.56437624
[3,]	-1.8325970	4.3185637	-0.04588057	-1.85369880	-0.55793805
[4,]	1.7099635	0.1239277	-2.03105844	0.37685888	11.06556521
[5,]	0.6216665	3.4822937	-0.72892670	-0.30797531	-0.34263686
[6,]	0.1504718	192.3431240	-1.61328707	1.22937850	0.70879566
[7,]	-1.2580457	0.5361613	1.20981230	0.64294858	-0.01998232
[8,]	0.7813287	1.0224632	2.71046180	-0.07719628	-8.83025591
[9,]	-1.1544210	-66.6994477	1.30846411	0.68260729	-14.20240591
[10,]	2.2993906	10.2349086	-0.14099909	-1.76940953	24.84998591
[11,]	-1.3825393	-8.3864444	-1.99433162	0.91545899	26.05482273

Lampiran 72. Weights DNN untuk Peramalan Inflow Rp 100.000,-

[[1]]					
[[1]] [[1]]					
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.35160598	0.24023195	-0.38917793	-1.47009406	-0.2670295
[2,]	-0.71198541	0.15321626	-0.44724564	-1.72796986	-2.5238421
[3,]	0.78665983	-0.77937559	-0.02113885	7.77410903	5.5630464
[4,]	1.64519552	-0.34866250	-0.59941746	-1.28629520	2.9655255
[5,]	-0.05781236	-6.31733906	-0.31086619	4.80112119	1.6066346
[6,]	0.28602518	2.17563162	-0.68163247	3.75253966	-2.4658459
[7,]	-1.39120430	2.42930825	-0.20205572	4.70221797	-0.9953207
[8,]	0.65662193	4.76807388	1.11680033	-3.68085246	-1.1464466
[9,]	-0.03712701	-0.75289128	0.82599921	4.87935359	-1.1242600
[10,]	-1.13039720	-0.18326095	1.25091898	-0.49248712	0.6881530
[11,]	1.80985109	-0.33805727	2.60809809	0.04793793	1.0129925
[12,]	-1.09659974	-0.08296338	-0.05103996	4.61140494	2.3374662
[13,]	0.65320671	2.83700103	2.22719419	1.38612980	-1.4432595
[14,]	-0.28656649	0.19825981	-0.01387212	-0.24176417	3.9109081
[15,]	-0.59955464	1.18739962	-1.54739970	1.62819202	-1.0043887
[16,]	0.13660517	3.05325898	-1.37988911	-11.44971066	0.2524345
[17,]	1.78967743	3.78356482	1.47987074	-7.13001460	2.2027745
[18,]	-1.01237298	0.11212472	-0.25492194	5.38175325	-61.6233179
[19,]	-0.27928182	-1.37060846	-0.32628038	2.80161738	-100.0082051
[20,]	2.31741296	1.08871137	-0.72663867	1.73655790	-2.2778363
[21,]	0.62874901	2.55390148	-1.95234865	-1.61061785	1.4929217
[22,]	-1.29491794	2.56286118	0.42294004	0.66488915	2.7183062
[23,]	2.11486448	2.30398567	1.18168479	-6.51487885	1.7974242
[24,]	-1.13815834	-5.75638161	0.91795938	1.41538654	-7.1159652
[25,]	0.29038081	3.56476704	0.09546755	-12.27126633	1.8352316
[26,]	-1.26973591	-0.51366276	-1.68443179	0.13028103	-7.5740245
[27,]	0.06191844	7.59368972	0.99032935	-9.42016792	-3.2130838

[28,]	2.25420426	-13.93707877	-0.70783178	11.15060957	2.4998454
[29,]	1.39342626	-7.66968902	-0.59402917	-8.81527687	47.8413993
[30,]	-0.56033236	3.58276461	-1.06589703	-3.69336683	8.1630586
[31,]	-0.67145938	-1.18246196	-0.94076703	-31.49423225	1.0340171
[32,]	0.59243855	-61.09494276	0.09391043	-4.82906372	5.9046780
[33,]	-1.17939052	43.09372228	0.06625338	-40.03233253	1.9073457
[34,]	-1.05871745	-80.93883397	-0.99065612	-0.84677600	39.0725912
[35,]	1.13790261	2.72438848	-0.07211499	17.54573700	0.2056151
[1,]	[.6]	[.7]			
[1,]	6.547150e-01	-0.19807490			
[2,]	-1.279512e+00	0.03803649			
[3,]	9.106624e-01	-0.65708108			
[4,]	9.645462e-02	-0.30950324			
[5,]	3.172835e-01	-1.35457858			
[6,]	-8.881533e-01	0.19033365			
[7,]	1.019213e+00	-0.18034468			
[8,]	-9.119167e-01	-0.78680050			
[9,]	-1.396097e+00	-0.86548084			
[10,]	1.646040e+00	-0.10023742			
[11,]	-4.421489e-03	0.04202599			
[12,]	-4.111717e+00	0.27258531			
[13,]	-4.458831e+00	6.36466474			
[14,]	-2.960984e-02	1.71719015			
[15,]	-8.669569e-02	-0.11771742			
[16,]	-4.251974e+00	-2.46890754			
[17,]	-2.806047e+00	36.34770135			
[18,]	9.591218e-01	-2.45184698			
[19,]	1.316349e+00	-3.10847031			
[20,]	-2.173366e-01	0.02870883			
[21,]	2.669899e+00	32.88555837			
[22,]	-1.048928e+00	2.74551621			
[23,]	-5.656058e-01	-0.34912947			
[24,]	1.078228e+00	-1.32826527			
[25,]	-2.860056e+00	0.43496818			
[26,]	-1.077336e+00	1.96268271			
[27,]	6.825034e+00	4.28515668			
[28,]	1.051663e+00	-4.60273728			
[29,]	-1.035951e+02	-2.98419438			
[30,]	-1.353061e+00	-4.32896572			
[31,]	-1.756797e+00	32.66296307			
[32,]	-5.580635e+00	-2.78481086			
[33,]	4.522068e-01	-6.66524933			
[34,]	-7.638552e-01	-1.35140534			
[35,]	-2.270130e+00	-2.48111307			
[[1]][[2]]					
[1,]	[.1]	[.2]	[.3]	[.4]	[.5]
[1,]	0.2682954	-0.2921051	1.2650306	-0.1695848	0.7736515
[2,]	0.1531522	1.5594124	0.2383972	-1.7879408	-0.2159089
[3,]	-0.1049508	-3.1763271	-0.6828203	-0.1856223	-2.5880825
[4,]	-0.9567502	1.6489981	1.1722963	-0.2373695	0.7588105
[5,]	-0.2452923	3.1579922	2.9015656	6.3836565	-1.1637315
[6,]	-0.6164504	-2.4974071	2.6263514	3.6186552	7.5665190
[7,]	-113.5280929	-5.3250216	-35.1863425	-114.3712967	18.1186946
[8,]	-5.3312478	-0.7111059	-1.5227830	0.3456227	0.8385641
[1,]	[.6]	[.7]	[.8]	[.9]	[.10]
[1,]	-0.8054817	-0.71152923	0.1719360	-0.3964277	0.57428924
[2,]	-0.3976633	0.12425992	-1.0711379	-1.4012220	-0.70326667
[3,]	-2.0577540	0.64580315	2.4041507	-80.6799174	-5.22904715
[4,]	-0.3015802	0.17484764	-0.6295068	1.1884082	-0.77698765
[5,]	1.7105281	0.76194102	-0.2976801	0.8425261	1.23386981
[6,]	1.0628382	-0.48543988	0.0473256	0.3413390	0.04693267
[7,]	-3.9138748	-0.05709542	-249.2852079	-0.3808139	-1.27632789
[8,]	-3.0615295	1.08786023	-0.8047087	-1.4913417	-2.02938999
[[1]][[3]]					
[1,]	[.1]				
[1,]	-0.83193326				
[2,]	-2.6998261				
[3,]	1.0710116				
[4,]	-0.8353639				
[5,]	0.7637270				
[6,]	0.9963509				
[7,]	3.2857084				

[8,]	-1.7265384
[9,]	0.9309110
[10,]	3.2317546
[11,]	1.0283312

Lampiran 73. *Weights* DNN untuk Peramalan *Inflow* Rp 50.000,-

[1,]	[1,]	[2,]	[3,]	[4,]
[1,]	0.164489516	7.826758e-01	0.18651143	-1.60064719
[2,]	0.357344034	-2.308583e-01	-0.28794542	-1.83851751
[3,]	0.512616381	7.506085e-01	0.28914191	-0.05223458
[4,]	0.022005605	-3.558720e+00	2.32400840	-0.09516393
[5,]	0.186483559	2.752810e+00	1.57906571	0.62509631
[6,]	0.045179533	1.196054e+00	0.03072450	-0.75888543
[7,]	0.778396969	2.092658e+00	1.62650722	-0.72558946
[8,]	-0.036007099	1.137195e+00	-4.28674612	-0.62044125
[9,]	0.025227014	9.586437e-01	4.35478807	0.05755500
[10,]	0.044586472	-2.489920e+00	-1.86672373	-1.29173802
[11,]	-0.552135509	2.938757e+00	-2.60738546	1.02221384
[12,]	-0.020752629	-6.352842e-03	-0.32221534	-1.22700447
[13,]	-36.790440942	6.276050e-01	-0.33348625	-0.65018743
[14,]	-0.073079121	-3.903810e+00	0.05923238	1.38196032
[15,]	-28.307731994	-1.960727e-01	-0.19068451	1.36227551
[16,]	0.003852508	-5.871204e-03	0.18395338	1.12694932
[17,]	0.796183939	-1.568576e+00	1.33303733	2.27760069
[18,]	0.762186861	-2.391453e+02	0.53130624	1.74015834
[19,]	1.809911725	-6.994461e+00	5.09604172	0.87251322
[20,]	1.381888108	-7.094162e+02	-705.84881975	-0.61535332
[21,]	2.177905617	1.420425e+00	8.78641259	-0.20757686
[22,]	2.886080709	2.538760e+00	-702.74181089	4.19071088
[23,]	2.021767321	1.572597e+00	12.56054015	1.08685746
[24,]	1.256773076	5.248192e+00	-302.94841940	7.00813843
[25,]	0.438405380	-6.890515e+00	11.91286565	-0.67962657
[26,]	0.514601270	-3.682403e+00	-4.46398146	0.81861321
[27,]	0.721962543	4.791544e-01	0.54842041	-0.85455634
[28,]	4.435578969	-7.574844e-01	-28.98673854	4.80155691
[29,]	0.016429181	8.751656e-01	14.28678268	0.51924942
[30,]	-0.135242295	-7.247725e+02	6.54607480	3.22563215
[31,]	0.993845616	-7.206349e+02	-443.13908134	-5.59817759
[32,]	1.330719850	-5.603843e+02	31.34441591	-2.44416681
[33,]	-0.543442554	1.289990e+00	-2.02952458	6.34562187
[34,]	-0.638770798	-6.680131e-01	15.88259582	-2.44628570
[35,]	-2.318624990	6.319065e-01	-3.07469321	6.02353733
[36,]	-1.148487394	-6.764907e+00	-510.03145995	-0.38485609
[1,]	[5,]	[6,]	[7,]	
[1,]	-1.63975187	-0.722602074	-0.4272605	
[2,]	0.90485309	0.139255350	0.6816381	
[3,]	-1.83382168	1.621538415	-0.9696601	
[4,]	-0.42053688	-0.530397573	-0.6980579	
[5,]	-2.35592589	2.224698075	1.7162595	
[6,]	2.16575342	-2.719231706	0.3196023	
[7,]	-4.20920573	-0.099732856	0.6348964	
[8,]	2.21544041	1.472621192	-2.8798603	
[9,]	3.28126237	13.098445447	1.0546533	
[10,]	-0.62422283	-1.598872297	-0.4146465	
[11,]	-0.08260738	5.511421963	-0.6691816	
[12,]	0.05555478	0.298158482	-0.5785989	
[13,]	-0.30109348	0.987828103	4.2886820	
[14,]	-0.25503861	-2.678477358	-0.4981148	
[15,]	-1.84023018	-0.153418492	-0.2878582	
[16,]	0.96166598	-1.026170127	0.5823195	
[17,]	14.08534925	-3.592841766	22.1588748	
[18,]	40.72176794	-13.199181296	2.2065501	
[19,]	-10.16539918	-13.883464220	-0.1165210	
[20,]	-5.11425831	-4.978534722	-0.7133778	
[21,]	-8.76992100	-11.735420591	-0.2738868	
[22,]	-0.95116957	14.677547121	0.1208054	
[23,]	0.66826539	-1.250751125	-1.1968714	
[24,]	11.81029691	-12.933614392	1.9741718	
[25,]	9.33470743	-15.170255064	-3.7741991	
[26,]	91.43785903	56.944469089	3.9870355	

[27,]	60.97334869	-7.141790629	5.1343177
[28,]	-0.40277944	-2.919826253	7.7218347
[29,]	46.89775480	-65.379614890	8.4956597
[30,]	28.8052143	-27.492873105	3.0571699
[31,]	39.23241374	26.211646211	1.9678658
[32,]	-10.58459392	0.003245866	-28.5554148
[33,]	-22.06734168	-1.197686770	-4.7093368
[34,]	20.45378591	0.767970870	-5.6089307
[35,]	-2.13628810	-10.254098417	-4.8722993
[36,]	-1.37132165	-8.956771566	-89.1606261

[1,]	[2,]	[3,]	
[1,]	-0.08840999	-1.6823339	1.4133353
[2,]	-12.20368518	-15.1222487	-0.4161507
[3,]	3.33104210	-0.1221778	-2.4300255
[4,]	2.41222680	-9.1998403	3.4699451
[5,]	-3.37412020	5.7480189	3.3803962
[6,]	-1.04525626	38.4933348	-0.6147773
[7,]	1.52682165	0.1304047	-2.2928236
[8,]	2.22906914	-21.3387309	-4.0794495

[1,]	[3,]
[1,]	0.4188634
[2,]	11.5883074
[3,]	-0.7552237
[4,]	-2.7041713

Lampiran 74. Weights DNN untuk Peramalan Inflow Rp 20.000,-

[1,]	[2,]	[3,]	[4,]	[5,]	[6,]	
[1,]	-0.47230780	-0.82646375	0.03783733	1.4562164	-0.66312320	-5.470936e-01
[2,]	3.46975261	2.24943127	0.07010133	-3.3589567	2.15162161	2.695028e-01
[3,]	2.15179877	-0.29700828	1.7129847	-2.00525473	7.051924e-02	
[4,]	0.27755479	-0.67460168	-1.74785389	-1.2324874	2.20145167	-1.734758e+00
[5,]	0.68528962	0.77197575	-0.95818824	-2.6614398	2.65448662	-2.848964e-03
[6,]	-2.39152419	2.74331126	-6.50733244	-1.7460603	5.40685535	2.538161e-01
[7,]	2.12566412	1.42339208	0.15037147	0.1221232	-0.14014116	8.171157e-01
[8,]	-0.27343327	2.66892782	1.48370195	-0.8154431	-2.79025131	3.907628e-01
[9,]	-0.28811582	-0.19045097	0.85238895	1.1980420	-4.89732888	1.033156e+00
[10,]	-0.06550160	0.00357457	-0.01278857	0.8980149	1.10019157	-1.873491e+00
[11,]	2.18009931	5.79974607	-1.05115846	-3.8001552	5.35918897	4.213153e-01
[12,]	-0.67126758	-1.38515502	4.42265592	0.8269970	-0.19827051	-8.015141e+00
[13,]	-0.42227132	-0.25016466	0.14889507	0.1515370	4.40352672	1.642845e+00
[14,]	0.03760548	-0.02737711	-0.20951848	-1.1731551	-1.08137846	-1.264669e+01
[15,]	0.18886858	-4.71641814	2.15263324	7.9884927	2.19218249	3.263722e+00
[16,]	-2.49145233	-1.36613374	1.46688587	13.6103194	-0.11896875	1.258147e+00
[17,]	-2.29080326	1.04858293	-2.16407598	9.4617421	4.22710341	-7.046447e+02
[18,]	-1.02658155	-0.64820238	1.69426815	-7.3628580	0.92885433	-1.884346e+02
[19,]	-712.18522575	0.03436446	0.71020868	9.4770338	0.17992671	-2.242410e-01
[20,]	0.67613248	-0.01838248	1.24798405	-5.4003704	0.37768700	-7.052632e+02
[21,]	-1.03777401	-0.75787365	2.16624471	10.6249115	-0.93240867	-8.331874e-01
[22,]	4.20190483	2.62982005	-2.06740087	-9.4146782	-1.23092122	-7.058027e+02
[23,]	-0.90688966	3.11833689	-3.23265499	9.4576296	1.54496365	9.352847e-01
[24,]	-2.64306617	2.29923407	-1.95791899	7.0774470	-0.06711119	2.352572e+00
[25,]	1.06581461	0.85964711	2.28688275	-14.3607338	1.01837948	3.318190e+00
[26,]	1.92123374	-0.23543858	4.27785738	12.2071151	-2.47027465	-2.121529e+00
[27,]	5.13622452	-11.06757916	8.83057340	1.3860845	0.11667590	1.668788e+02
[28,]	7.86193453	-1.11597765	9.94218023	6.1593493	-0.01386736	1.211155e-01
[29,]	9.47838853	34.81281587	10.22708133	-61.9767146	-96.37285071	-4.269257e+02
[30,]	-701.14168844	-1.03981021	2.45161874	-4.6941477	0.49955208	-7.039237e+02
[31,]	1.84234530	2.00203817	-709.27708472	-1.0049237	-135.73274458	2.950611e+00
[32,]	5.20578746	-5.67093622	2.30173699	-14.2218389	-0.80222567	2.119873e+00
[33,]	8.95448434	-11.89184297	-439.91367091	-119.2384268	-132.59148177	-5.467834e+02
[34,]	11.30789304	-7.15379362	-704.46980068	-9.0638735	-0.41769304	2.036962e+00

[1,]	[2,]	[3,]	[4,]	[5,]	[6,]	[7,]	
[1,]	0.6801625	-0.7599185	-0.03717142	-0.3650128	1.497015	-0.2624505	-0.9746295
[2,]	4.9164808	-17.8923104	-0.66349663	-10.0687991	-43.854495	2.8762145	-0.3650510
[3,]	-6.9236896	21.4393677	3.64352664	1.5137724	-39.838409	2.5276807	0.3735157
[4,]	-2.9306562	6.3869447	-1.16529636	-0.7521932	36.188144	-2.6312486	-1.1344173
[5,]	-1.2520678	-4.8483372	0.24391857	-0.7719040	-2.706525	1.1286103	0.5417224
[6,]	0.4555460	-2.2628687	1.03921682	0.7396164	-2.544048	0.3256092	1.0797753
[7,]	0.1590353	-217.0120314	8.92858782	-0.5326460	-47.747755	4.5683649	5.2717805
[8,]							
[1,]	5.933238	-1.6338039					
[2,]	-1.270434	-12.2744155					

```
[3,] -14.255703 1.1331091
[4,] -8.818213 1.1869956
[5,] 7.974406 0.6050823
[6,] 3.556940 -0.4544063
[7,] 6.582956 -2.2340299
```

```
[[1]][[3]]
[1,]
[1,] -0.8311720
[2,] -1.0995459
[3,] -0.9012421
[4,] 1.2116995
[5,] -1.3486274
[6,] -0.7647306
[7,] 0.9130002
[8,] 1.4571176
[9,] 0.8061203
[10,] -0.9932162
```

Lampiran 75. Weights DNN untuk Peramalan *Inflow* Rp 10.000,-

```
[[1]]
[[1]][[1]]
[1,] [1,] [2,]
[1,] -1.786659123 1.976235e+00
[2,] 0.330301161 1.087982e-01
[3,] 0.280136488 9.502878e-02
[4,] 0.122019406 -2.354077e-01
[5,] -0.271226080 3.410813e-02
[6,] 0.002146582 -1.057675e-01
[7,] -1.637828861 -9.654008e-01
[8,] 0.360821909 -1.480258e+01
[9,] 0.427015395 1.457104e-01
[10,] -0.008896122 -1.414651e+01
[11,] 0.081212712 -5.973583e-01
[12,] -0.178897774 -5.908452e-01
[13,] -0.184086551 -4.674726e-01
[14,] -0.358736795 -2.769347e-01
[15,] -0.445096235 -2.213717e-01
[16,] -0.628446001 -1.892879e-01
[17,] -0.623358450 -2.034554e-01
[18,] -0.324963626 -3.345812e-01
[19,] -0.365397870 -2.187908e-01
[20,] -0.088832331 7.261511e-03
[21,] -0.079059005 -1.783968e+00
[22,] -0.473874441 -8.721826e-01
[23,] 0.127843596 8.537623e-01
[24,] -0.116023647 1.019164e+00
[25,] -0.732225936 6.264603e+00
[26,] -14.763058735 -1.932385e+00
[27,] 0.146734608 1.923693e-01
[28,] 0.227832163 2.824081e-01
[29,] 0.626644086 -7.011921e+02
[30,] 1.063520867 4.378183e-01
[[1]][[2]]
[1,] [1,] [2,] [3,] [4,] [5,]
[1,] -1.801218 -0.06081242 -2.112475 -0.418445 -0.354563
[2,] 2.717363 1.63564976 3.148285 -1.398894 1.564268
[3,] -2.106897 -2.21371948 -2.324817 2.245225 -2.307325
```

```
[[1]][[3]]
[1,]
[1,] -1.0894855
[2,] 6.5981411
[3,] 0.7866104
[4,] 7.8082192
[5,] -7.2959116
[6,] 1.3906319
```

Lampiran 76. *Weights* DNN untuk Peramalan *Inflow* Rp 5.000,-

[[1]]					
[[1]][[1]]					
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1.057521e-01	-2.747433e-01	0.05932544	0.18372580	0.19213915
[2,]	-4.866711e-01	5.429819e-01	-0.40168913	-0.69063675	-1.38046644
[3,]	3.461647e+00	-4.149280e-01	-0.07152276	0.34710212	0.51355018
[4,]	7.507590e-01	-4.706481e+00	0.84405190	-0.07352236	-1.20448235
[5,]	3.693060e+00	-3.587078e+00	1.06284223	-0.17692487	-3.06352307
[6,]	4.889462e+00	-1.153470e+00	1.05282813	-0.54546432	2.69190964
[7,]	-4.773625e+00	6.028369e-01	-0.23752568	-0.21680175	1.86428496
[8,]	-1.917623e-01	6.596100e-02	-0.59150624	0.01790728	1.25091198
[9,]	-1.016775e+01	3.111311e+00	-0.33759953	-4.08763315	1.87629597
[10,]	1.123962e+00	8.364932e-03	1.00965607	-1.62617323	12.59276897
[11,]	-4.415584e-02	3.720316e+00	0.61024094	0.49641440	-1.36711436
[12,]	-6.886328e-03	-8.831138e-02	0.58588931	0.87641269	13.88753190
[13,]	6.471094e+00	3.315800e+00	-2.07978629	-1.27461081	0.74182918
[14,]	7.979749e+00	-1.644226e+00	-0.82312297	-1.69080498	0.05006802
[15,]	7.682785e+00	-4.293386e-01	-0.67206003	-2.37391614	-1.06187841
[16,]	1.311670e+01	3.307195e+01	-0.35723458	-2.91398481	1.24200040
[17,]	1.347358e+01	3.199747e+01	-0.02287508	-3.67981174	0.86840263
[18,]	1.360175e+01	4.468173e+01	0.29612575	-3.99546270	1.15786692
[19,]	-7.028802e+02	1.798651e+01	1.41483814	-4.18749547	-1.53348877
[20,]	4.401642e+00	3.989514e+00	1.08277626	-10.03732963	-107.15807568
[21,]	1.251298e+01	-6.628218e-01	0.78492291	40.43353341	0.68669438
[22,]	-7.064618e+02	-4.271276e+00	-0.64659051	2.89585926	3.70795729
[23,]	2.865388e+00	1.018182e+02	-2.60679948	0.07308709	-1.51646546
[24,]	1.024088e+01	3.821034e+00	-1.48428228	-2.38755958	-0.48512073
[25,]	1.019983e+01	2.158562e+00	1.24127781	1.56825725	-148.98920337
[26,]	6.993976e+00	2.176461e+00	2.95034532	-7.97527696	3.98856514
[27,]	6.763147e+00	7.230753e+00	59.79127975	0.78252666	-23.26277840
[28,]	-6.961075e+02	2.851730e+01	59.09331905	-732.66959118	43.35192298
[29,]	1.584463e+00	-1.014132e+02	0.67223904	-9.09226037	-96.50603817
[30,]	-7.009565e+00	-5.913329e-01	-0.93685691	31.36127888	-867.66198979
[31,]	-6.495810e+00	-7.070857e+02	-710.61760700	10.06097395	-2.09707753
[32,]	4.608020e+00	-1.413428e+00	-1.94407768	2.13380222	-0.79758991
[[1]][[2]]					
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1.095584	-1.196394	-0.5235817	-0.5789004	33.549045
[2,]	3.297812	14.734880	1574.5972962	-2.5973932	-355.208985
[3,]	-1.851895	-66.736767	3.0958810	-127.4574857	43.679918
[4,]	-3.314130	-4198.457307	-1.2996772	-1.6203338	368.672340
[5,]	1.047559	264.828051	-0.5941828	1.4223308	-125.690980
[6,]	-2.128935	-22.995860	-0.3534093	-2.1042211	-4.351298
[[1]][[3]]					
	[,1]				
[1,]	-0.05209398				
[2,]	6.84261882				
[3,]	0.32280005				
[4,]	-0.75440458				
[5,]	-16.36708505				
[6,]	-0.49336804				

Lampiran 77. *Weights* DNN untuk Peramalan *Inflow* Rp 2.000,-

[[1]]					
[[1]][[1]]					
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.820087332	0.55488800	1.80648616	-0.202681219	1.31295366
[2,]	0.488572973	3.07940271	3.36637719	-0.468943604	1.06513685
[3,]	6.073956903	2.01020291	-3.26336195	0.3046694975	0.02459804
[4,]	-3.155484125	0.08444248	1.53310533	1.518390852	-0.86531666
[5,]	-0.000764407	-0.01593519	-0.04135177	1.763154956	1.32319983
[6,]	0.132420284	-0.77667527	0.15816476	-1.183606388	0.13885842
[7,]	0.707954729	0.53759616	-3.45952917	-1.276457254	1.79263159
[8,]	-0.239698024	-1.72597978	-0.81999511	-1.512670795	-0.43064097
[9,]	1.984473937	-1.80045802	-7.59839364	-1.171104899	-0.96222958
[10,]	2.255002996	-1.22717783	0.12572085	1.959357077	0.53757952
[11,]	2.077378508	-0.11414624	1.13668200	0.007645872	-0.66958599
[12,]	1.401977910	-709.09389006	1.40712941	-0.842615198	0.63880561

[13,]	39.586516873	0.39687834	-1.19657174	-0.601160105	-1.72398983
[14,]	3.769119479	1.64973617	2.70233958	1.074459406	-1.74243008
[15,]	3.063182050	-706.15810970	3.55107832	0.260597835	0.68980417
[16,]	5.150290643	5.23550497	4.50898639	-0.314271980	0.33096318
[17,]	-2.689363595	1.84929354	4.92630764	-0.749630095	0.87106771
[18,]	-15.460819845	1.44235671	-2.91486525	-0.952198330	-1.93424558
[19,]	0.287632062	2.20538270	-4.15292605	2.048040303	1.13757910
[20,]	-0.560668031	-2.97103196	-1.17770002	0.839920078	1.12549470
[21,]	-0.983737936	-0.70489939	2.70927431	2.008687116	0.95706833
[22,]	1.567555568	-6.05412127	4.07175959	-0.421373572	0.78641026
[23,]	3.686819997	-0.30346238	2.63724351	-0.440834423	2.19207351
[24,]	2.267382670	-3.15810808	5.06573664	-1.027380598	-1.45380985
[25,]	5.350472069	-1.17594740	-0.14969566	-0.250519127	-0.58310385
[26,]	-10.885933743	16.29664957	-6.09672012	0.471859466	0.40972398
[27,]	3.027631858	-2.33879091	0.67407478	1.358939821	-0.88198164
[28,]	-11.653490498	-4.29773377	-10.74901801	0.564168603	0.08555044
[29,]	15.837592245	-1.23998496	-22.52642140	0.455980090	0.74624317

[1,]	0.2353009	-0.90466867	0.34909061	-0.09019765	0.3832703
[2,]	1.8484520	-0.65218385	3.05848803	0.64852798	-3.5013266
[3,]	0.5112265	-0.26245464	1.20613863	0.31042320	-2.3906704
[4,]	-2.6156567	-0.93466284	3.43599823	-4.91916681	-1.1280236
[5,]	-0.1011725	0.82116121	-0.22030575	-0.03745866	-0.9376024
[6,]	0.9758906	-1.62425917	0.59946362	0.48271439	1.9311743
[7,]	-6.6319367	-1.03040367	0.79018999	0.06550310	1.3424790
[8,]	0.7221918	-1.26192931	1.02624620	-0.58783979	0.5274387
[9,]	-7.3966235	0.39218463	0.08654512	-0.11402065	0.8228519
[10,]	-321.8950423	-1.13143826	-686.39913787	5.44150330	84.3839176
[11,]	-80.7641762	0.54414448	20.02198868	3.05814457	86.4631594
[12,]	9.1846734	1.17660893	21.27836234	-54.34605584	86.3806278
[13,]	4.9444348	0.02522857	18.55652382	2.15592555	87.4229116
[14,]	-361.7587176	0.51513317	26.23499391	-3.28707831	-88.4683727
[15,]	11.8332639	-0.65410976	-476.59617632	-56.71039874	88.3595170
[16,]	7.1342174	0.50364199	25.65244032	-1.47525924	86.7242233
[17,]	1.9489376	-1.27211922	5.23281268	2.06722002	36.1468262
[18,]	-517.1764108	-0.07677115	14.35593505	-434.98209265	21.6678795
[19,]	-296.4875461	-1.34531938	17.38141474	4.83000746	-0.7375463
[20,]	5.4050042	-0.26631756	13.82632151	6.02375532	-1.2355022
[21,]	7.2322931	1.08756300	-689.38062702	6.83563768	-0.2371313
[22,]	-3.1387393	0.70056780	-5.29363710	-5.68362166	6.4269357
[23,]	4.3964074	-0.44275951	-4.87211033	-5.78295172	-11.0142864
[24,]	8.3358295	-0.78851997	-2.78637402	5.78295172	-13.1088357
[25,]	-8.4019380	-0.85677571	1.84978030	-43.09791568	37.4157466
[26,]	12.1536249	-0.74641901	4.68234079	4.98972064	-1.3517211
[27,]	-2.0421423	0.29798362	5.25317814	2.88717441	-0.3843536
[28,]	-62.8082855	-1.01955220	-22.90928158	0.92917474	87.7538728
[29,]	-0.4952821	2.87089742	-7.06013848	-0.16830933	15.4647356

[[1]] [[2]]

[1,]	-1.3975040	-1.6817831	-0.8224129	0.71087267	0.63126532
[2,]	0.3205176	7.3878074	-1.5546041	3.21525609	-1.17650987
[3,]	-1.1755018	-1.6551017	-1.4295363	-7.51734073	1.73021994
[4,]	0.7300553	4.1755338	0.9936844	0.73816343	-4.76515540
[5,]	1.2192217	-0.5757482	0.8968621	2.88142170	1.09538835
[6,]	-0.6226475	-1.1714397	-0.2026660	2.47560941	0.05053853
[7,]	-1.2990164	-0.6774890	-1.5514355	-4.38226339	0.54304154
[8,]	-2.3659897	-0.1248367	-0.5209727	1.29433780	-0.00989832
[9,]	0.1941206	-0.7621336	3.6068940	0.01860073	1.54820977
[10,]	-0.7828139	-1.1547862	0.6017708	0.43171540	-0.97044700
[11,]	1.8999538	1.1852903	2.8701028	-6.93949358	0.13155449

[1,]	-0.70570630	-1.1821557	1.4111973	0.39851588	6.3530504
[2,]	-0.29582139	1.4162879	-2.4508109	-0.03978335	-2.9537022
[3,]	-1.37495361	-1.2476402	2.1714865	2.30727254	-19.5204003
[4,]	2.13424764	1.8039021	-3.2223586	-0.67814441	7.4425787
[5,]	-0.07148051	1.3144537	2.5226013	0.38404395	6.3956930
[6,]	1.00138538	1.1823482	-0.5657468	-0.47462529	5.4074306
[7,]	-3.05126177	-1.9624955	-1.2596992	2.91186288	6.9036741
[8,]	-0.82428163	-0.7386531	0.2034191	-1.33387072	4.5383861
[9,]	-0.43319755	-2.1056046	56.1068304	0.61752753	-5.5690980
[10,]	1.76564500	-1.4920435	0.8739355	0.49525412	0.8937124
[11,]	2.77139570	0.9711046	0.9510881	-0.83709173	-15.7932475

[[1]][[3]]	
	[,1]
[1,]	0.02585344
[2,]	-2.47855690
[3,]	-1.04649591
[4,]	-0.27442054
[5,]	1.09763805
[6,]	0.68378012
[7,]	-0.46337961
[8,]	-2.21510062
[9,]	1.38792939
[10,]	0.24293583
[11,]	0.30100455

Lampiran 78. Weights DNN untuk Peramalan *Inflow* Rp 1.000,-

[[1]]		[[1]]		[[1]]		[[1]]		[[1]]		[[1]]	
	[,1]		[,2]		[,3]		[,4]		[,5]		[,6]
[1,]	0.31874617	-2.316786567	0.88092221	-4.41308833	-1.80443306						
[2,]	0.11574746	-0.552808097	-0.93042725	-0.99626557	-0.81074008						
[3,]	-1.07133055	-3.909853418	-0.54844550	-2.37923740	-2.13156292						
[4,]	-0.29916772	0.634507631	0.28149645	-0.42467163	0.97931540						
[5,]	0.59454513	-0.539075511	-0.98993624	1.99176590	0.88575503						
[6,]	0.68979430	1.577780479	1.67270628	-3.12336422	1.43086921						
[7,]	-0.90807618	-1.656383337	-0.97982702	0.56483424	0.67100717						
[8,]	-0.06367602	0.020215583	0.06961218	-1.21483017	0.04619276						
[9,]	-1.32667268	-0.938908359	-0.55163864	0.93806135	2.22434304						
[10,]	0.04352161	0.084493687	2.02258634	1.62944049	0.06400790						
[11,]	1.40177950	-0.003852364	1.09282805	0.04598618	-0.48441672						
[12,]	1.05578151	51.478705816	0.27733849	3.28153796	1.62479223						
[13,]	-0.23823356	-0.776779073	0.71289428	0.94188621	2.35756575						
[14,]	1.28744470	-5.179436787	1.21660083	0.23563122	0.20046854						
[15,]	0.74139013	0.237233489	0.44024186	1.15091897	0.36545715						
[16,]	0.08934727	-2.344173587	0.10883062	-0.99692558	1.92840037						
[17,]	-0.95494386	-2.315246335	0.85976414	-1.61182546	2.73528194						
[18,]	0.10484962	1.336634028	2.37051966	-5.37008605	-705.66849287						
[19,]	1.12552126	1.274985332	-0.68394507	-5.46611565	2.42950859						
[20,]	0.68297852	-2.013595130	0.22604417	34.57249602	-659.46313853						
[21,]	-0.49631064	0.670743721	-0.01635467	-1.02514437	1.46314437						
[22,]	-2.08528684	-0.510103046	0.10851766	-3.60743919	-671.59370048						
[23,]	-0.67486594	-1.499389424	0.46954478	-0.92757312	0.02435858						
[24,]	-2.11906119	-0.018123765	1.45534832	-5.98307926	-3.06032250						
[25,]	-1.26519802	-1.681934430	0.79495735	-4.81526720	-666.54741639						
[26,]	-0.37366156	-2.029694396	-1.11964511	33.95655375	-84.79790760						
[27,]	-0.68755543	2.461050829	-1.80667725	-5.51330339	-1.72070760						
[28,]	-0.87215883	-1.069366304	0.89292590	0.91137436	-36.61112712						
[29,]	-0.10176101	2.266619828	0.44816796	-3.55913123	-209.91358361						
[30,]	-0.25378053	1.453218460	1.22702839	-0.64611538	-12.80807136						
[31,]	-1.85374045	0.393967638	-0.76180434	-0.15982066	-54.57683792						
[32,]	-0.07794607	-1.428383143	0.41937541	1.77908711	3.57450690						
[33,]	0.96856634	-2.323936501	-1.23994336	2.43123651	-1.28003729						
[34,]	0.18492596	-1.091580491	0.71157397	4.10363140	-475.05970915						
[35,]	-1.37994358	-0.952127294	-0.63321301	2.83148915	21.71112392						
	[,6]		[,7]		[,8]		[,9]		[,10]		[,11]
[1,]	1.27718138	0.19184495	-0.19379614	0.005465303	2.69643445						
[2,]	0.22828932	4.24000099	1.12259618	-0.087087516	2.24321779						
[3,]	-0.30881306	-0.74657763	-0.27895461	-1.732787430	-0.52833763						
[4,]	0.95982913	0.08520145	0.44174864	0.940622482	2.62648799						
[5,]	0.54882237	0.45822455	0.66656609	-0.188388886	4.5701371						
[6,]	0.42551309	-0.62010332	0.70552363	-1.130939789	1.75534519						
[7,]	0.64350004	0.74892454	0.98608764	0.991502725	1.65794490						
[8,]	-1.36030614	-1.98692958	-2.24137625	0.486064457	1.53387906						
[9,]	-0.19850611	-0.41574318	-2.59517046	0.27549933	0.53072443						
[10,]	0.61930268	-0.75062126	0.84814338	1.709962539	0.29556636						
[11,]	2.06820961	-0.28326175	1.06778955	-0.121191082	0.99560157						
[12,]	-0.30528475	-43.58495446	-0.09424252	0.240272749	0.46812549						
[13,]	0.28124561	0.74904565	-1.16210885	-0.213012516	-0.24807823						
[14,]	0.69131734	-44.12129360	-1.07263290	0.044184959	0.72231080						
[15,]	0.04636144	0.25341247	-1.01378749	0.485239591	0.24446730						
[16,]	0.11302936	1.59151792	-0.52691535	3.105802426	0.58072117						
[17,]	0.99533187	4.01611939	2.33655199	0.464783323	1.03561986						
[18,]	-0.68115136	-1.65618040	-2.30887832	-1.082011420	-1.63234239						

[19,]	-1.27705725	-4.77043639	-2.64242365	-1.005396299	-0.38065550
[20,]	-1.46869775	2.57644833	3.09199294	2.512518714	0.09435481
[21,]	-0.31347407	3.10383404	4.33045573	0.226048265	-0.14425178
[22,]	-1.70365949	1.93038293	0.91967832	-20.375702494	0.09758718
[23,]	-1.35051466	2.70213804	-2.63261685	-3.625588867	0.24612392
[24,]	-1.10209368	1.60942997	-1.20410528	-0.118885366	0.46640705
[25,]	-1.09954301	2.88243629	1.00754389	-17.743485108	0.75479348
[26,]	1.21551378	30.47632005	-1.45307968	0.618862511	1.43544588
[27,]	0.33087649	-4.41962148	-3.98486995	0.173932760	0.08513338
[28,]	1.39027512	1.06822747	-4.13000235	0.313771333	-0.25879293
[29,]	0.87204699	0.06697623	-1.04404469	-2.846671581	0.82347802
[30,]	-1.08081702	0.15692839	2.66059485	18.533969356	-0.62399509
[31,]	0.49582159	25.19141070	-1.21297466	16.110986150	1.81941870
[32,]	1.05262756	-23.98602179	-1.31435869	1.043756830	-0.61667341
[33,]	-1.27464995	34.88189893	-1.09879383	0.916509439	0.18454106
[34,]	-0.19366673	3.77168492	-2.64827769	15.114213311	2.21452131
[35,]	-1.29508364	-0.06338048	-0.86847508	2.141052018	-0.92856737

[[1]] [[2]]

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-1.6617712	0.9548601	-0.002469886	-0.06015684	-0.08406008
[2,]	0.4961627	-1.1712064	-0.065935919	-0.95329322	-0.11213061
[3,]	-3.7560499	-1.1747051	0.760862926	-1.99914642	0.52006025
[4,]	0.3472901	-0.7399648	2.582984165	-0.18637130	1.98483526
[5,]	0.8609773	-1.5674391	1.084392081	-0.29128160	-0.26434460
[6,]	-1.1245081	-1.5295729	1.136287698	-0.33064415	2.73179910
[7,]	0.4261954	-0.5305024	-1.122488242	0.35105679	-0.22958739
[8,]	2.7965763	2.5892986	-0.021578588	-0.23787152	-2.10454783
[9,]	-1.4001946	-0.7168285	-0.963732931	-1.24003058	5.75816787
[10,]	1.9788944	1.2309554	-0.702413608	1.78610289	-2.62777622
[11,]	0.3196466	0.6288124	1.093829854	-0.63677943	-0.88848948

	[,6]	[,7]
[1,]	-1.88575941	-1.3026254
[2,]	-1.87931840	0.1045209
[3,]	-0.16549767	0.7979759
[4,]	1.26851407	-0.7647178
[5,]	-0.10691026	1.3567684
[6,]	-0.72310535	2.9882756
[7,]	0.05350638	-0.1237685
[8,]	-11.23207422	0.1150603
[9,]	-142.28646850	1.4819820
[10,]	4.55232531	-0.1695409
[11,]	-2.93070401	-0.2354792

[[1]] [[3]]

	[,1]
[1,]	0.7286154
[2,]	1.1301828
[3,]	1.9694705
[4,]	-0.6245923
[5,]	1.5777596
[6,]	-0.4828101
[7,]	3.4494568
[8,]	-1.2057108

Lampiran 79. Model Ramalan Manual ARIMAX *Outflow* Rp 100.000,-

$$\begin{aligned}
 Y_{1,t} = & (280, 61t + 18941, 9D_1 + 12308, 4D_2 - 147, 60tD_1 - \\
 & 147, 35tD_2 - 17528, 9M_{1,t} - 16331, 7M_{2,t} - 18424, 9M_{3,t} - \\
 & 15692, 5M_{4,t} - 15242, 3M_{5,t} - 12726, 1M_{6,t} - 11210, 6M_{7,t} - \\
 & 16193, 9M_{8,t} - 12917, 8M_{9,t} - 15518, 6M_{10,t} - 23686M_{11,t} - \\
 & 8195, 5M_{12,t} + 18198L_{1,t-1} + 3354, 6L_{2,t-1} - 4043, 8L_{3,t-1} + \\
 & 1272, 7L_{4,t-1} + 2815, 7L_{1,t} + 11505, 2L_{2,t} + 16131, 4L_{3,t} + 17060, 8L_{4,t}) \\
 & (1 + 0, 18B^{10} - 0, 81B^{12}) + (1 + 0, 20B^2)a_t - 0, 18Y_{1,t-10} + 0, 81Y_{1,t-12} \\
 = & 280, 61t + 51, 9(t - 10) - 228, 7(t - 12) \\
 & + 18941, 9D_{1,t} + 3504, 2D_{1,t-10} - 15437, 6D_{1,t-12} \\
 & + 12308, 4D_{2,t} + 2277D_{2,t-10} - 10031, 3D_{2,t-12} \\
 & - 147, 60tD_{1,t} - 27, 3(t - 10)D_{1,t-10} + 120, 3(t - 12)D_{1,t-12} \\
 & - 147, 35tD_{2,t} - 27, 2(t - 10)D_{2,t-10} + 120, 1(t - 12)D_{2,t-12} \\
 & - 17528, 9M_{1,t} - 3242, 8M_{1,t-10} + 14286M_{1,t-12} \\
 & - 16331, 7M_{2,t} - 3021, 4M_{2,t-10} + 13310, 2M_{2,t-12} \\
 & - 18424, 9M_{3,t} - 3408, 6M_{3,t-10} + 15016, 3M_{3,t-12} \\
 & - 15692, 5M_{4,t} - 2903, 11M_{4,t-10} + 12789, 4M_{4,t-12} \\
 & - 15242, 3M_{5,t} - 2819, 8M_{5,t-10} + 12422, 5M_{5,t-12} \\
 & - 12726, 1M_{6,t} - 2354, 3M_{6,t-10} + 10371, 8M_{6,t-12} \\
 & - 11210, 6M_{7,t} - 2073, 9M_{7,t-10} + 9136, 6M_{7,t-12} \\
 & - 16193, 9M_{8,t} - 2995, 9M_{8,t-10} + 13198M_{8,t-12} \\
 & - 12917, 8M_{9,t} - 2389, 8M_{9,t-10} + 10528M_{9,t-12} \\
 & - 15518, 6M_{10,t} - 2870, 9M_{10,t-10} + 12647, 7M_{10,t-12} \\
 & - 23686M_{11,t} - 4381, 9M_{11,t-10} + 19304M_{11,t-12} \\
 & - 8195, 5M_{12,t} - 1516, 17M_{12,t-10} + 6679, 3M_{12,t-12} \\
 & + 18198L_{1,t-1} + 3366, 6L_{1,t-11} - 14831, 4L_{1,t-13} \\
 & + 3354, 6L_{2,t-1} + 620, 6L_{2,t-11} - 2734L_{2,t-13} \\
 & - 4043, 8L_{3,t-1} - 748, 1L_{3,t-11} + 3295, 7L_{3,t-13} \\
 & + 1272, 7L_{4,t-1} + 235, 4L_{4,t-11} - 1037, 2L_{4,t-13} \\
 & + 2815, 7L_{1,t} + 520, 9L_{1,t-10} - 2294, 8L_{1,t-12} \\
 & + 11505, 2L_{2,t} + 2128, 5L_{2,t-10} - 9376, 7L_{2,t-12} \\
 & + 16131, 4L_{3,t} + 2984, 3L_{3,t-10} - 13147, 1L_{3,t-12} \\
 & + 17060, 8L_{4,t} + 3156, 25L_{4,t-10} - 13904, 6L_{4,t-12} \\
 & + a_t + 0, 20a_{t-1} - 0, 18Y_{1,t-10} + 0, 81Y_{1,t-12}
 \end{aligned}$$

Lampiran 80. Uji Non-Linieritas (*White Test*) Secara Simultan

Data	Replikasi										Proporsi Non-Linier
	1	2	3	4	5	6	7	8	9	10	
DNN-1											
$Y_{1,t}$	0.072	0.037	0.001	0.466	0.100	0.066	0.159	0.007	0.576	0.031	0.4
$Y_{2,t}$	0.164	0.227	0.603	0.117	0.076	0.666	0.036	0.544	0.815	0.003	0.2
$Y_{3,t}$	0.002	0.001	0.000	0.014	0.027	0.024	0.590	0.006	0.000	0.610	0.8
$Y_{4,t}$	0.589	0.055	0.318	0.000	0.008	0.079	0.067	0.321	0.361	0.530	0.2
$Y_{5,t}$	0.099	0.184	0.001	0.472	0.043	0.140	0.004	0.038	0.071	0.599	0.4
$Y_{6,t}$	0.760	0.760	0.170	0.170	0.210	0.730	0.580	0.550	0.240	0.340	0.0
$Y_{7,t}$	0.388	0.533	0.011	0.671	0.000	0.449	0.065	0.152	0.770	0.252	0.2
$Y_{8,t}$	0.541	0.013	0.019	0.022	0.000	0.047	0.241	0.940	0.060	0.000	0.6
$Y_{9,t}$	0.001	0.180	0.020	0.380	0.680	0.011	0.100	0.000	0.950	0.060	0.4
$Y_{10,t}$	0.884	0.077	0.500	0.005	0.011	0.725	0.788	0.504	0.099	0.172	0.2
$Y_{11,t}$	0.002	0.670	0.018	0.000	0.310	0.180	0.000	0.001	0.290	0.018	0.6
$Y_{12,t}$	0.040	0.287	0.001	0.003	0.014	0.015	0.082	0.949	0.714	0.015	0.6
$Y_{13,t}$	0.617	0.065	0.710	0.004	0.003	0.003	0.045	0.003	0.002	0.829	0.6
$Y_{14,t}$	0.004	0.031	0.840	0.570	0.001	0.000	0.056	0.001	0.000	0.550	0.6
DNN-2											
$Y_{1,t}$	0.000	0.000	0.012	0.000	0.510	0.150	0.027	0.000	0.490	0.001	0.7
$Y_{2,t}$	0.130	0.031	0.000	0.081	0.603	0.063	0.014	0.552	0.145	0.195	0.3
$Y_{3,t}$	0.020	0.017	0.026	0.024	0.026	0.023	0.017	0.020	0.020	0.027	1.0
$Y_{4,t}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.0
$Y_{5,t}$	0.005	0.006	0.043	0.400	0.580	0.000	0.000	0.000	0.044	0.010	0.8
$Y_{6,t}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.0
$Y_{7,t}$	0.000	0.000	0.001	0.003	0.001	0.087	0.009	0.004	0.000	0.140	0.8
$Y_{8,t}$	0.046	0.002	0.000	0.001	0.440	0.000	0.000	0.027	0.040	0.000	0.9
$Y_{9,t}$	0.270	0.990	0.410	0.130	0.350	0.500	0.067	0.930	0.000	0.120	0.1
$Y_{10,t}$	0.001	0.029	0.320	0.570	0.260	0.300	0.160	0.044	0.055	0.000	0.4
$Y_{11,t}$	0.310	0.680	0.590	0.530	0.660	0.630	0.800	0.890	0.830	0.730	0.0
$Y_{12,t}$	0.000	0.017	0.000	0.000	0.430	0.001	0.012	0.000	0.001	0.000	0.9
$Y_{13,t}$	0.300	0.990	0.870	0.210	0.460	0.310	0.910	0.470	0.380	0.200	0.0
$Y_{14,t}$	0.052	0.444	0.002	0.035	0.415	0.756	0.286	0.063	0.275	0.002	0.3
SSA-DNN											
$Y_{1,t}$	0.000	0.009	0.000	0.005	0.056	0.000	0.000	0.039	0.004	0.000	0.9
$Y_{2,t}$	0.660	0.360	0.760	0.870	0.850	0.930	0.930	0.780	0.790	0.670	0.0
$Y_{3,t}$	0.670	0.770	0.430	0.610	0.750	0.850	0.990	0.740	0.640	0.960	0.0
$Y_{4,t}$	0.410	0.330	0.360	0.350	0.370	0.310	0.360	0.420	0.320	0.320	0.0
$Y_{5,t}$	0.000	0.000	0.000	0.490	0.480	0.000	0.720	0.000	0.000	0.000	0.7
$Y_{6,t}$	0.634	0.237	0.161	0.393	0.155	0.638	0.325	0.133	0.019	0.086	0.1
$Y_{7,t}$	0.047	0.008	0.001	0.066	0.399	0.003	0.130	0.100	0.000	0.739	0.5
$Y_{8,t}$	0.434	0.910	0.016	0.846	0.394	0.008	0.041	0.002	0.002	0.050	0.5
$Y_{9,t}$	0.601	0.025	0.085	0.682	0.297	0.099	0.138	0.581	0.344	0.392	0.1
$Y_{10,t}$	0.941	0.158	0.081	0.014	0.490	0.324	0.860	0.086	0.040	0.368	0.2
$Y_{11,t}$	0.232	0.017	0.964	0.076	0.647	0.015	0.935	0.678	0.706	0.038	0.3
$Y_{12,t}$	0.643	0.015	0.716	0.189	0.031	0.158	0.176	0.074	0.321	0.472	0.2
$Y_{13,t}$	0.410	0.770	0.560	0.760	0.610	0.570	0.590	0.550	0.520	0.560	0.0
$Y_{14,t}$	0.203	0.197	0.093	0.282	0.117	0.863	0.628	0.037	0.343	0.192	0.1

Lampiran 81. Syntax ARIMAX *Outflow* Rp 100.000,-

```

data outflow;
input y t d1 d2 td1 td2 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 L1b L2b L3b
L4b L1 L2 L3 L4 ;
datalines;
5275.02758      1      1      0      1      0      1
      0      0      0      0      0      0      0
      0      0      0      0      0      0      0
      0      0      0      0      0      0      0
4274.4599 2      1      0      2      0      0      1
      0      0      0      0      0      0      0
      0      0      0      0      0      0      0
      0      0      0      0      0      0      0
4867.83275      3      1      0      3      0      0
      0      1      0      0      0      0      0
      0      0      0      0      0      0      0
      0      0      0      0      0      0      0
\* dan seterusnya sampai data baris ke 180
.      180      0      0      0      0      0      0
      0      0      0      0      0      0      0
      0      0      1      0      0      0      0
      0      0      0      0      0      0      0
;

proc arima data =outflow;
    identify var=y crosscorr=(t d1 d2 td1 td2 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10
M11 M12 L1b L2b L3b L4b L1 L2 L3 L4) nlag=24 noprint;
    run;
    estimate p=(10,12) q=(2) input=(t d1 d2 td1 td2 M1 M2 M3 M4 M5 M6 M7 M8 M9
M10 M11 M12 L1b L2b L3b L4b L1 L2 L3 L4) noconstant method=cls;
    forecast out = outflow lead=24 printall;
    run;
    proc univariate data=outflow normal;
        var residual;
    run;

proc export data=work.outflow
    outfile='E:\TA-LJ\Output\ARIMAX Forecast\Forecast O_100.xls'
    dbms=excel
    replace;
run;

```

Lampiran 82. Syntax DNN *Outflow* Rp 100.000,-

```
library(neuralnet)
library(svMisc)
library(grid)
library(MASS)
RAW=read.table("D:/DNN2/Yt.csv",header=TRUE,sep=";")
Data=RAW[,1]
nfull=length(Data)
training=Data[1:(nfull-12)]
testing=Data[(nfull-11):nfull]
n=length(training)

Dummy=read.csv("D:/DNN2/Dummy.csv",header=TRUE,sep=";")
DummyO=cbind(Dummy[,3:19],Dummy[,20:27])
DummyI=cbind(Dummy[,3:19],Dummy[,24:31])
DummyO2=DummyO[79:length(Dummy[,1]),]
DummyI2=DummyI[80:length(Dummy[,1]),]

DummyDLNN=DummyO
H1=c(1:10)
H2=c(1:10)
nreplikasi=10
SET=NULL
STEPMAX=1e+5

PACF=matrix(0,24,3)
temp=pacf(training,lag.max=24) #Identifikasi Input Berdasarkan PACF yang Signifikan
for (i in 1:24){
  PACF[i,1]=i
  PACF[i,2]=temp[i,1]$acf
  PACF[i,3]=if(abs(PACF[i,2])>(1.96/sqrt(n)))) 1 else 0
}
nfore=12
Lag=subset(PACF,PACF[,3]==1)[,1]
#Lag=c(10,12)
p=length(Lag)
xbar=mean(Data)
stdev=sd(Data)
Data_std=(Data-xbar)/stdev
Zt=Data_std[1:n]

a=max(Lag)+1
ZtOutput=Zt[a:n]
DummyTrain=DummyDLNN[a:n,]
ZtLag=matrix(0,n-max(Lag),p)
colnames(ZtLag)=c(paste(rep("Lag.",length(Lag)),Lag,sep=""))

for (i in 1:p){
```

```

        awal=max(Lag)-Lag[i]+1
        akhir=n-Lag[i]
        ZtLag[,i]=Zt[awal:akhir]}

Hidden=cbind(rep(H1,times=length(H2)),rep(H2,each=length(H1)))
N=length(Hidden[,1])
input=(cbind(ZtLag,DummyTrain))

FitsModel=list()
ForecastModel=list()
DLNNModel=list()
SummaryDLNN=list()

for ( i in 1:N){
  h1=Hidden[i,1]
  h2=Hidden[i,2]
  DLNN=list()
  Fits=list()
  Forecast=list()
  SummaryReplikasi=list()

  for (k in 1:nreplikasi){
    delayedAssign("do.next", {next})
    if (k==nreplikasi) {
      SummaryReplikasi[[k]]=c( k , 1e+9 , 1e+9 , 1e+9 , 1e+9 )
      Fits[[k]]=Forecast[[k]]=DLNN[[k]]=c(0)
    }
  }
  tryCatch({
    if (SET=="NULL") {
      set.seed(k)
    } else {
      set.seed(SET)
    }
  },error = function(e) force(do.next),
  warning = function(w) force(do.next))
  #===== Menghitung Forecast=====
  YtTest=c(Zt,rep(0,nfore))
  for (l in (n+1):(n+nfore)){
    DummyTest=DummyDLNN[l,]
    ZtLagTest=matrix(0,1,p)
    colnames(ZtLagTest)=c(paste(rep("Lag.",length(Lag)),Lag,sep=""))

    for (j in 1:p){
      ZtLagTest[,j]=YtTest[(l-Lag[j])]}
    inputTest=cbind(ZtLagTest,DummyTest)
    YtTest[l]=compute(DLNN[[k]],covariate=inputTest)$net.result
  }
}

```

```

Fits[[k]]=unlist(DLNN[[k]]$net.result)*stdev+xbar
NtTrain=(ZtOutput*stdev+xbar)-(unlist(DLNN[[k]]$net.result)*stdev+xbar)
RMSETrain=sqrt(sum(NtTrain^2)/length(NtTrain))
pembilang=2*abs(NtTrain)
penyebut=(ZtOutput*stdev+xbar)+(unlist(DLNN[[k]]$net.result)*stdev+xbar)
sMAPETrain=sum(pembilang/penyebut)/length(NtTrain)*100

Forecast[[k]]=YtTest[(n+1):(n+nfore)]*stdev+xbar
NtTest=testing-(YtTest[(n+1):(n+nfore)]*stdev+xbar)
RMSETest=sqrt(sum(NtTest^2)/length(NtTest))
pembilang=2*abs(NtTest)
penyebut=testing+Forecast[[k]]
sMAPETest=sum(pembilang/penyebut)/length(NtTest)*100

SummaryReplikasi[[k]]=c(k,RMSETrain,sMAPETrain,RMSETest,sMAPETest)
}
ListReplikasi=t(matrix(unlist(SummaryReplikasi), nrow = 5, byrow = FALSE))
colnames(ListReplikasi)=c("k", "RMSE_Train", "sMAPE_Train", "RMSE_Test", "sMAPE_Test")
best_k=ListReplikasi[which.min(ListReplikasi[,4]),1] #Memilih replikasi terbaik ke-k
SummaryDLNN[[i]]=c(i,h1,h2,subset(as.data.frame(ListReplikasi),k==best_k))
FitsModel[[i]]=Fits[[best_k]]
ForecastModel[[i]]=Forecast[[best_k]]
DLNNModel[[i]]=DLNN[[best_k]]

progress(i)
if (i == N) cat("Done!\n")
}

##
ListModel=t(matrix(unlist(SummaryDLNN), nrow = length(SummaryDLNN[[1]]),
byrow = FALSE))
colnames(ListModel)=c("i", "Neuron_h1", "Neuron_h2", "Replikasi_ke", "RMSE_Train", "sMAPE_Train", "RMSE_Test", "sMAPE_Test")
B=ListModel[which.min(ListModel[,7]),1]
plot(DLNNModel[[B]])
Actual=ZtOutput*stdev+xbar
win.graph()
par(mfrow=c(1,2),mar=c(2.7,3,1.2,0.4),oma=c(0,0,0,0))
par(mgp=c(1.9,0.5,0))
ts.plot(c(Actual,rep(NA,nfore)),lwd=2,ylab="DLNN Training")
lines(c(FitsModel[[B]],ForecastModel[[B]]),col="red",lwd=1.5)
testingplot=as.matrix(cbind(testing,ForecastModel[[B]]))
matplot(testingplot,type="l",lwd=2,col=c(1,2),lty=c(1,1),ylab="DLNN Testing")
BestForecast=cbind(seq(1,nfull),Data,c(rep(NA,max(Lag)),FitsModel[[B]],ForecastModel[[B]]))
row.names(BestForecast)=NULL
colnames(BestForecast)=c("t","Aktual","Forecast")

```

Lampiran 83. Syntax SSA-DNN *Outflow* Rp 100.000,-

```
library(Rssa)
library(neuralnet)
RawDataSSA<-read.csv("E:/Dimas/Yt.csv",header=TRUE,sep=";")
DataSSA=RawDataSSA[,1]
nfull=length(DataSSA)

DummySSA=read.csv("E:/Dimas/DummySSA.csv",header=TRUE,sep=";")
DummyO=cbind(DummySSA[,1:17],DummySSA[,18:25],DummySSA[,30:41])
DummyI=cbind(DummySSA[,1:17],DummySSA[,22:29],DummySSA[,30:41])
DummyO2=DummyO[79:nfull,]
DummyI2=DummyI[80:nfull,]
      DTrain=DummyO[1:(nfull-12),]
      DTest=DummyO[(nfull-11):nfull,]

DummyDLNN=DummyO[,1:(ncol(DummyO)-12)]
r=18
H1=c(1:10)
H2=c(1:10)
nreplikasi=2
SET="NULL"
STEPMAX=1e+5
#===== SSA =====#

SSAtrain=DataSSA[1:(nfull-12)]
SSAtest=DataSSA[(nfull-11):nfull]
      SSA<-ssa(SSAtrain)
      eigen=SSA$sigma
      selisih=c(NA,diff(eigen,1)*-1)
      View(cbind(eigen,selisih))
      win.graph(7,3.5)
      plot(SSA,main=NULL,mar=c(2.7,3,1.2,0.4))

win.graph()
plot(SSA,type="series",groups=as.list(1:r))
plot(SSA,type="paired",idx=1:r,plot.contrib=FALSE)
parestimate(SSA, list(2:3), method = "esprit")$periods
parestimate(SSA, list(9:10), method = "esprit")$periods
recon<-reconstruct(SSA,groups=list(Trend=c(1),Seasonality=c(2:r)))
win.graph()
par(mfrow=c(3,1),mar=c(2.7,2.9,1.2,0.4))
ts.plot(recon$Trend,ylab="Trend")
ts.plot(recon$Seasonality,ylab="Seasonal")
ts.plot(residuals(recon),ylab="Noise")
      YtTrend<-recon$Trend
      YtRegresi<-recon$Seasonal

win.graph(7,3.5)
par(mfrow=c(2,2),mar=c(2.7,3,1.2,0.4),oma=c(0,0,0,0))
par(mgp=c(1.9,0.5,0))
```

```

#===== TREND ANALYSIS =====#
t=seq(1,length(DataSSA))
T=cbind(t,t^2,t^3)
TTrain=as.data.frame(T[1:(nfull-12),])
TTest=as.data.frame(T[(nfull-11):nfull,])
Trend=lm(YtTrend~.,data=TTrain)
      TrendFore=predict(Trend,TTest)
      ts.plot(YtTrend,ylab="Trend",lwd=2)
      lines(Trend$fitted.values,col="red",lwd=2)
#===== Regresi Dummy untuk komponen Seasonal hasil SSA =====#
fReg=as.formula(paste("YtRegresi ~", paste(names(DTrain),-1, collapse= "+")))
      Regresi<-lm(fReg,data=DTrain)
      RegresiFore=predict(Regresi,DTest)
      ts.plot(YtRegresi,ylab="Seasonal",lwd=2)
      lines(Regresi$fitted.values,col="red",lwd=2)
NoiseTrend=YtTrend-Trend$fitted.values
NoiseSeasonal=YtRegresi-Regresi$fitted.values
Resi=residuals(recon)+NoiseTrend+NoiseSeasonal
ts.plot(residuals(recon),ylab="Noise SSA",lwd=2)
ts.plot(Resi,ylab="Noise SSA + Residuals",lwd=2)

#===== DNN =====#
library(neuralnet)
library(svMisc)
Data=Resi
testingDLNN=SSAtest-(TrendFore+RegresiFore)
n=length(Data)
PACF=matrix(0,24,3)
temp=pacf(Resi,lag.max=24,plot=FALSE)
for (i in 1:24){
  PACF[i,1]=i
  PACF[i,2]=temp[i,1]$acf
  PACF[i,3]=if(abs(PACF[i,2])>(1.96/sqrt(n))) 1 else 0
}

nfore=12
Lag=subset(PACF,PACF[,3]==1)[,1]
p=length(Lag)
xbar=mean(Data)
stdev=sd(Data)
Data_std=(Data-xbar)/stdev
Zt=Data_std[1:n]

a=max(Lag)+1
ZtOutput=Zt[a:n]
DummyTrain=DummyDLNN[a:n,]
ZtLag=matrix(0,n-max(Lag),p)
colnames(ZtLag)=c(paste(rep("Lag.",length(Lag)),Lag,sep=""))

```



```

for (i in 1:p){
  awal=max(Lag)-Lag[i]+1
  akhir=n-Lag[i]
  ZtLag[,i]=Zt[awal:akhir]}
Hidden=cbind(rep(H1,times=length(H2)),rep(H2,each=length(H1)))
N=length(Hidden[,1])
input=(cbind(ZtLag,DummyTrain))
###
FitsModel=list()
ForecastModel=list()
DLNNModel=list()
SummaryDLNN=list()
for ( i in 1:N){
  h1=Hidden[i,1]
  h2=Hidden[i,2]
  ##
    DLNN=list()
    Fits=list()
    Forecast=list()
    SummaryReplikasi=list()
    for (k in 1:nreplikasi){
      delayedAssign("do.next", {next})
    if (k==nreplikasi) {
      SummaryReplikasi[[k]]=c( k , 1e+9 , 1e+9 , 1e+9 , 1e+9 )
      Fits[[k]]=Forecast[[k]]=DLNN[[k]]=c(0)
    }
    tryCatch({
      if (SET=="NULL"){
        set.seed(k)
      } else {
        set.seed(SET)}
      f=as.formula(paste("ZtOutput ~ ", paste(names(input), collapse= "+")))
      DLNN[[k]]=neuralnet(f,data=input,hidden=c(h1,h2),linear.output=TRUE,act.f
ct="logistic",likelihood=TRUE,stepmax=STEPMAX)
    },error = function(e) force(do.next),
      warning = function(w) force(do.next))
    #===== Menghitung Forecast =====#
    YtTest=c(Zt,rep(0,nfore))
      for (l in (n+1):(n+nfore)){
        DummyTest=DummyDLNN[l,]
        ZtLagTest=matrix(0,1,p)
        colnames(ZtLagTest)=c(paste(rep("Lag.",length(Lag)),Lag,sep=""))
          for (j in 1:p){
            ZtLagTest[,j]=YtTest[(l-Lag[j])]}
        inputTest=cbind(ZtLagTest,DummyTest)
        YtTest[l]=compute(DLNN[[k]],covariate=inputTest)$net.result

      }

```

```

Fits[[k]]=unlist(DLNN[[k]]$net.result)*stdev+xbar
  NtTrain=(ZtOutput*stdev+xbar)-(unlist(DLNN[[k]]$net.result)*stdev+xbar)
  RMSETrain=sqrt(sum(NtTrain^2)/length(NtTrain))
  pembilang=2*abs(NtTrain)
  penyebut=(ZtOutput*stdev+xbar)+(unlist(DLNN[[k]]$net.result)*stdev+xbar)
  sMAPETrain=sum(pembilang/penyebut)/length(NtTrain)*100

  Forecast[[k]]=YtTest[(n+1):(n+nfore)]*stdev+xbar
  NtTest=testingDLNN-(YtTest[(n+1):(n+nfore)]*stdev+xbar)
  RMSETest=sqrt(sum(NtTest^2)/length(NtTest))
  pembilang=2*abs(NtTest)
  penyebut=testingDLNN+Forecast[[k]]
  sMAPETest=sum(pembilang/penyebut)/length(NtTest)*100
SummaryReplikasi[[k]]=c(k,RMSETrain,sMAPETrain,RMSETest,sMAPETest)
}#kurung tutup untuk replikasi ke k

##
ListReplikasi=t(matrix(unlist(SummaryReplikasi), nrow = 5, byrow = FALSE))
colnames(ListReplikasi)=c("k", "RMSE_Train", "sMAPE_Train", "RMSE_Test", "sMAPE_Test")
best_k=ListReplikasi[which.min(ListReplikasi[,4]),1]
SummaryDLNN[[i]]=cbind(i,h1,h2,subset(as.data.frame(ListReplikasi),k==best_k))
FitsModel[[i]]=Fits[[best_k]]
ForecastModel[[i]]=Forecast[[best_k]]
DLNNModel[[i]]=DLNN[[best_k]]
progress(i)
if (i == N) cat("Done!\n")

}
###
ListModel=t(matrix(unlist(SummaryDLNN), nrow = 8, byrow = FALSE))
colnames(ListModel)=c("i", "Neuron_h1", "Neuron_h2", "Replikasi_ke", "RMSE_Train", "sMAPE_Train", "RMSE_Test", "sMAPE_Test")
B=ListModel[which.min(ListModel[,7]),1]
win.graph()
inputplot=as.matrix(cbind(testingDLNN,Forecast[[B]]))
matplot(inputplot,type="l",col=c(1,2),lty=c(1,1),lwd=2)
  FitsBestDLNN=unlist(DLNNModel[[B]]$net.result)*stdev+xbar
  ForecastBestDLNN=unlist(ForecastModel[[B]])

#=== PERHITUNGAN SSA_DLNN AKHIR ===
FitsSSA_DLNN=Trend$fitted.values+Regresi$fitted.values+c(rep(0,a-1),FitsBestDLNN)
ForecastSSA_DLNN=TrendFore+RegresiFore+ForecastBestDLNN
  NtTrain=SSAtrain-FitsSSA_DLNN
  RMSETrain=sqrt(sum(NtTrain^2)/length(NtTrain))
  pembilang=2*abs(NtTrain)
  penyebut=SSAtrain+FitsSSA_DLNN
  sMAPETrain=sum(pembilang/penyebut)/length(NtTrain)*100

  NtTest=SSAtest-ForecastSSA_DLNN

```

```

RMSETest=sqrt(sum(NtTest^2)/length(NtTest))
pembilang=2*abs(NtTest)
penyebut=SSAtest+FitsSSA_DLNN
sMAPETest=sum(pembilang/penyebut)/length(NtTest)*100
SummarySSA_DLNN=c(h1,h2,ListModel[B,4],RMSETrain,sMAPETrain,RMSETest,sMAPETest)
BestSSA_DLNN=data.frame(t(sapply(SummarySSA_DLNN,c)))
colnames(BestSSA_DLNN)=c("Neuron_h1","Neuron_h2","k","RMSE_Train",
"sMAPE_Train","RMSE_Test","sMAPE_Test")
win.graph()
par(mfrow=c(1,2),mar=c(2.7,3,1.2,0.4),oma=c(0,0,0,0))
par(mgp=c(1.9,0.5,0))
trainingplot=as.matrix(cbind(SSAtrain,FitsSSA_DLNN))
matplot(trainingplot,type="l",lwd=2,col=c(1,2),lty=c(1,1),ylab="SSA-DLNN
Training")
testingplot=as.matrix(cbind(SSAtest,ForecastSSA_DLNN))
matplot(testingplot,type="l",lwd=2,col=c(1,2),lty=c(1,1),ylab="SSA-DLNN
Testing")

win.graph()
matplot(as.matrix(cbind(SSAtest,TrendFore+RegresiFore)),col=c(1,2),lty=c(1,1),lwd=2,
type="l")
Hasil=cbind(DataSSA,c(Trend$fitted.values,TrendFore),c(Regresi$fitted.values,RegresiFore),
c(c(rep(0,a-1),FitsBestDLNN),ForecastBestDLNN),c(FitsSSA_DLNN,ForecastSSA_DLNN))
row.names(Hasil)=NULL
colnames(Hasil)=c("Aktual","Trend.Analysis","Regresi.Dummy","DLNN","SSA-DLNN")

save(DLNNModel,file="DLNN Model dari SSA-DLNN O_100.rda")
write.csv(Hasil,"E:/Dimas/SSA-DLNN O_100.csv")
write.csv(ListModel,"E:/Dimas/List DLNN dari SSA-DLNN O_100.csv")
write.table(Hasil,"E:/Dimas/ Best SSA-DLNN O_100.txt",sep=";")

```

(Halaman ini sengaja dikosongkan)

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FMKSD ITS:

Nama : Dimas Ewin Ashari

NRP : 062116 4500 0006


menyatakan bahwa data yang digunakan dalam Tugas Akhir / Thesis ini merupakan data sekunder yang diambil dari ~~Penelitian~~ / ~~Buku~~ / Tugas Akhir / ~~Thesis~~ / ~~Publikasi lainnya~~ yaitu:

Sumber : Prilyandari Dina Saputri, Statistika FMKSD ITS

Keterangan : Data *Inflow* dan *Outflow* Nasional Periode Bulanan Menurut Pecahan Mata Uang Tahun 2003 -2016


Surat Pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui
Pembimbing Tugas Akhir



(Dr. Suhartono)
NIP. 19710929 199512 1001

Surabaya, 27 Juli 2018



(Dimas Ewin Ashari)
NRP. 062116 4500 0006

*(coret yang tidak perlu)

BIODATA PENULIS



Penulis dengan nama lengkap Dimas Ewin Ashari lahir di Kabupaten Tulungagung, pada 12 Maret 1995 sebagai anak kedua dari 2 bersaudara dari orangtua bernama Hariyanto dan Mariyati. Penulis mendapatkan pendidikan formal di TK Katholik St. Maria dan SDN Kenayan II Tulungagung, SMPN 1 Tulungagung, SMAN 1 Kauman, Tulungagung, Diploma III Statistika Institut Teknologi Sepuluh Nopember Surabaya melalui program beasiswa Bidikmisi dan melanjutkan pendidikan melalui program Lintas Jalur S1 Statistika ITS. Selama perkuliahan Lintas Jalur, penulis beberapa kali berkesempatan mengikuti kompetisi statistika tingkat nasional sebagai semifinalis dan perempat finalis pada NSC UB 2016, KSN IPB 2016 dan 2018, Dokter Data Undip 2018 hingga akhirnya mendapatkan Juara I pada *National Statistics Challenge* di UB Tahun 2017. Penulis juga berkesempatan menjadi Asisten Dosen Mata Kuliah Analisis Data Kategorik.

Segala kritik, saran atau diskusi lebih lanjut mengenai Tugas Akhir ini dapat disampaikan kepada penulis melalui alamat email dimasewin1995@gmail.com atau nomor telepon 085735212606.